



National Technical University of Athens  
School of Electrical & Computer Engineering  
Division of Computer Science

# **Out-of-distribution robustness in mission-critical computer vision applications**

**Ph.D. Thesis**

Anastasios Arsenos

Athens, July 2024





National Technical University of Athens  
School of Electrical & Computer Engineering  
Division of Computer Science  
Artificial Intelligence and Learning Systems Laboratory

**Out-of-distribution robustness in mission-critical computer  
vision applications**

Ph.D. Thesis  
of  
**Anastasios Arsenos**

Supervisor: **Prof. Stefanos Kollias**

Submitted in School of Electrical & Computer Engineering of  
National Technical University of Athens

Athens, July 2024







National Technical University of Athens  
School of Electrical & Computer Engineering  
Division of Computer Science  
Artificial Intelligence and Learning Systems Laboratory

## Out-of-distribution robustness in mission-critical computer vision applications

Ph.D. Thesis  
of  
**Anastasios Arsenos**

**Supervising Committee:** Stefanos Kollias  
Andreas-Georgios Stafylopatis  
Giorgos Stamou

Approved by the advisory committee on July 10th, 2024.

.....  
Stefanos Kollias      Andreas-Georgios Stafylopatis      Giorgos Stamou  
Professor N.T.U.A      Professor N.T.U.A      Professor N.T.U.A

.....  
Athanasios Voulodimos      Fotis Koumboulis  
Assistant Professor N.T.U.A      Professor N.K.U.A.

.....  
Dimitrios Kollias      George Matsopoulos  
Assistant Professor Queen Mary University      Professor N.T.U.A.

Athens, July 2024

The content of this Ph.D. Thesis does not reflect the official opinion of the National Technical University of Athens. Responsibility for the information and views expressed in this thesis lies entirely with the author.

Content that is reused from publications that the author has (co-)authored (excerpts, figures, tables, etc.) is under copyright with the respective paper publishers (IEEE, Elsevier, Springer etc) and is cited accordingly in the current text. Content that is reused from third-party publications appears with the appropriate copyright note. Reuse of any such content by any interested party requires the publishers' prior consent, according to the applicable copyright policies. Content that has not been published before is copyrighted jointly as follows:

Copyright ©Anastasios Arsenos, 2024  
Electrical & Computer Engineer N.T.U.A.  
All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών  
& Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών  
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Ευρωστία σε εκτός κατανομής δεδομένα για κρίσιμες  
εφαρμογές της όρασης υπολογιστών

Διδακτορική Διατριβή  
ΤΟΥ  
Αναστάσιου Π. Αρσένου

Επιβλέπων: Καθ. Στέφανος Κόλλιας

Υποβλήθηκε στη Σχολή Ηλεκτρολόγων Μηχανικών  
& Μηχανικών Υπολογιστών  
του Εθνικού Μετσόβιου Πολυτεχνείου

Αθήνα, Ιούλιος 2024





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών  
& Μηχανικών Υπολογιστών  
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών  
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

**Ευρωστία σε εκτός κατανομής δεδομένα για κρίσιμες  
εφαρμογές της όρασης υπολογιστών**

Διδακτορική Διατριβή  
του  
**Αναστάσιου Π. Αρσένου**

Συμβουλευτική Επιτροπή: Στέφανος Κόλλιας  
Ανδρέας-Γεώργιος Σταφυλοπάτης  
Γεώργιος Στάμου

Εγκρίθηκε από την πενταμελή εξεταστική επιτροπή την 10<sup>η</sup> Ιουλίου 2024.

.....  
Στέφανος Κόλλιας      Ανδρέας-Γεώργιος Σταφυλοπάτης      Γεώργιος Στάμου  
Καθηγητής ΕΜΠ                      Καθηγητής ΕΜΠ                      Καθηγητής ΕΜΠ

.....  
Αθανάσιος Βουλόδημος                      Φώτιος Κουμπουλής  
Επίκουρος Καθηγητής ΕΜΠ                      Καθηγητής ΕΚΠΑ

.....  
Δημήτριος Κόλλιας                      Γεώργιος Ματσόπουλος  
Επίκουρος Καθηγητής Queen Mary University                      Καθηγητής ΕΜΠ

Αθήνα, Ιούλιος 2024

.....

Αναστάσιος Π. Αρσένος

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright ©Αναστάσιος Π. Αρσένος, 2024.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

# Abstract

Artificial intelligence (AI) has progressed explosively in recent years. Driven by the advent of deep learning, AI is being used in a variety of applications, across multiple scientific fields, in industry as well as in medicine. Out-of-distribution (OOD) robustness is crucial in mission-critical computer vision applications because these scenarios often involve encountering unforeseen or novel situations that may differ significantly from the training data. In mission-critical contexts, such as autonomous vehicles, medical diagnosis, or security systems, the models need to make reliable and safe decisions. If the model encounters situations or inputs that fall outside the distribution it was trained on, it may provide inaccurate or unreliable predictions, leading to potentially dangerous consequences. Ensuring OOD robustness is essential to enhance the generalization capabilities of computer vision models, enabling them to handle diverse and unexpected scenarios in real-world applications. It helps prevent the system from making critical errors when faced with novel inputs, thereby improving safety, reliability, and performance in mission-critical tasks.

The emergence of Out-of-Distribution (OOD) robustness or Domain Generalization research has become a crucial tool for achieving reliable performance in medical imaging and autonomous driving. In the context of medical imaging, OOD robustness is vital because medical datasets can vary significantly due to differences in patient demographics, imaging equipment, and conditions. Researchers and practitioners recognize the need for models that can generalize well to diverse and previously unseen medical scenarios to ensure accurate diagnoses and treatment plans.

Similarly, in autonomous driving, OOD robustness is essential as driving conditions can be highly dynamic and unpredictable. Ensuring that self-

---

driving vehicles can handle unforeseen scenarios, such as adverse weather conditions, unusual environment configurations, or unexpected obstacles, is critical for their safe deployment in the wild. OOD robustness research in both medical imaging and autonomous driving aims to enhance the generalization capabilities of machine learning models, enabling them to perform reliably in real-world scenarios beyond the training distribution. This research contributes to the development of more trustworthy and resilient systems in these mission-critical domains.

This study proposes methodologies and advancements aimed at enhancing OOD robustness in mission-critical applications. From transfer learning techniques tailored for medical imaging to novel sensor configurations for UAV perception systems and state-of-the-art deep learning architectures for image recognition, significant progress has been made in addressing the challenges posed by OOD data. In the domain of medical imaging, we explored methodologies for enhancing the generalization capabilities of diagnostic models, considering factors such as data heterogeneity, limited sample sizes, and domain shifts across different healthcare facilities. For UAV sense and avoid systems, we investigated techniques for perceptual robustness to ensure safe operation in dynamic environments. In image recognition, we examined approaches for mitigating the impact of OOD data, such as adversarial training, domain generalisation, and uncertainty estimation, to enhance model reliability across diverse datasets and environmental conditions.

In summary, this PhD thesis highlights the critical importance of OOD robustness in mission-critical applications and underscores the need for continued research and innovation in this area. By synthesizing insights from diverse studies and identifying key challenges and advancements, this PhD thesis aims to contribute to the ongoing discourse on enhancing the reliability and safety of AI-driven systems in real-world scenarios. Through interdisciplinary collaboration and rigorous experimentation, we strive to develop effective solutions that ensure the resilience and efficacy of AI technologies across medical imaging, UAV sense and avoid systems, and image recognition domains.



---

**Keywords:** Out-of-Distribution Robustness, Mission-Critical Applications, Computer Vision, Deep Learning

---

# Περίληψη

Η τεχνητή νοημοσύνη (AI) έχει προχωρήσει εκρηκτικά τα τελευταία χρόνια. Η τεχνητή νοημοσύνη και η Βαθιά Μάθηση χρησιμοποιείται σε ποικίλες εφαρμογές, σε πολλά επιστημονικά πεδία, τόσο στη βιομηχανία όσο και στην ιατρική. Η ευρωστία σε εκτός κατανομής δεδομένα (OOD) είναι ζωτικής σημασίας σε εφαρμογές που είναι κρίσιμες για την αποστολή, επειδή αυτά τα σενάρια συχνά συνεπάγονται αντιμετώπιση μη-προβλεπόμενων ή νέων καταστάσεων που μπορεί να διαφέρουν σημαντικά από τα δεδομένα εκπαίδευσης. Σε κρίσιμα για την αποστολή πλαίσια, όπως αυτόνομα οχήματα, ιατρικά διάγνωση, ή συστήματα ασφαλείας, τα μοντέλα πρέπει να είναι αξιόπιστα και με ασφαλείς αποφάσεις. Εάν το μοντέλο συναντήσει καταστάσεις ή εισόδους που δεν εμπίπτουν στην κατανομή στην οποία εκπαιδεύτηκε, μπορεί να προκύψουν ανακριβείς ή αναξιόπιστες προβλέψεις, οδηγώντας σε δυνητικά επικίνδυνες συνέπειες. Εξασφάλιση OOD ευρωστίας είναι απαραίτητη για τη βελτίωση των δυνατοτήτων γενίκευσης του αλγορίθμου σε μοντέλα όρασης, που τους επιτρέπουν να χειρίζονται διαφορετικά και απροσδόκητα σενάρια σε πραγματικές εφαρμογές. Βοηθά να αποτρέψει το σύστημα από το να γίνουν κρίσιμα σφάλματα όταν αντιμετωπίζονται νέες εισροές, βελτιώνοντας έτσι την ασφάλεια, την αξιοπιστία, και απόδοση σε κρίσιμα περιβάλλοντα.

Η εμφάνιση της ευρωστίας εκτός διανομής (OOD) ή του Τομέα Γενίκευσης έχει γίνει ένα σημαντικό εργαλείο για την επίτευξη αξιόπιστης επιδόσης στην ιατρική απεικόνιση και την αυτόνομη οδήγηση. Στο πλαίσιο της ιατρικής απεικόνισης, η ευρωστία OOD είναι ζωτικής σημασίας επειδή τα ιατρικά σύνολα δεδομένων μπορεί να ποικίλλουν σημαντικά λόγω των διαφορών στα δημογραφικά στοιχεία των ασθενών και τον εξοπλισμό απεικόνισης. Οι ερευνητές και οι επαγγελματίες αναγνωρίζουν την ανάγκη για μοντέλα που μπορούν να γενικεύουν καλά σε ποικίλα και άγνωστα προηγουμένως ιατρικά σενάρια για την εξασφάλιση ακριβών διαγνώσεων και σχεδίων θεραπευ-

---

ίας.

Ομοίως, στην αυτόνομη οδήγηση, η ευρωστία OOD είναι απαραίτητη, καθώς οι συνθήκες οδήγησης μπορεί να είναι εξαιρετικά δυναμικές και απρόβλεπτες. Η διασφάλιση ότι τα αυτόνομα οχήματα μπορούν να χειριστούν απρόβλεπτα σενάρια, όπως αντίξοες καιρικές συνθήκες, ασυνήθιστες διαμορφώσεις περιβάλλοντος ή απροσδόκητα εμπόδια, είναι κρίσιμη για την ασφαλή ανάπτυξή τους. Η έρευνα ευρωστίας OOD τόσο στην ιατρική απεικόνιση όσο και στην αυτόνομη οδήγηση στοχεύει να ενισχύσει τις δυνατότητες γενίκευσης των μοντέλων μηχανικής μάθησης, επιτρέποντάς τους να αποδίδουν αξιόπιστα σε πραγματικά σενάρια πέρα από τη κατανομή εκπαίδευσης. Αυτή η έρευνα συμβάλλει στην ανάπτυξη πιο αξιόπιστων και ανθεκτικών συστημάτων σε αυτούς τους κρίσιμους για την αποστολή τομείς.

Στα πλαίσια της Διαδακτορικής Διατριβής αναπτύξαμε καινοτόμες μεθόδους που σχετίζονται με την ευρωστία OOD στην ιατρική απεικόνιση, τα συστήματα αίσθησης και αποφυγής UAV και την αναγνώριση εικόνας. Στον τομέα της ιατρικής απεικόνισης, αναπτύξαμε μεθοδολογίες για τη βελτίωση των δυνατοτήτων γενίκευσης των διαγνωστικών μοντέλων, λαμβάνοντας υπόψη παράγοντες όπως η ετερογένεια των δεδομένων, τα περιορισμένα μεγέθη δειγμάτων και οι μετατοπίσεις τομέα σε διαφορετικές εγκαταστάσεις υγειονομικής περίθαλψης. Για συστήματα αίσθησης και αποφυγής UAV, δημιουργήσαμε τεχνικές αντιληπτικής ευρωστίας για να διασφαλίσουμε την ασφαλή λειτουργία σε δυναμικά περιβάλλοντα. Στην αναγνώριση εικόνων, εξετάσαμε προσεγγίσεις για τον μετριασμό του αντίκτυπου των δεδομένων OOD, όπως η εκπαίδευση σε αντίθεση, η γενίκευση τομέα και η εκτίμηση αβεβαιότητας, για να ενισχύσουμε την αξιοπιστία του μοντέλου σε διάφορα σύνολα δεδομένων.

Μέσω αυτής της Διδακτορικής Διατριβής ανακαλύψαμε πολλές μεθοδολογίες και προόδους που στοχεύουν στην ενίσχυση της ευρωστίας OOD σε κρίσιμες εφαρμογές, από τεχνικές μεταφοράς μάθησης προσαρμοσμένες για ιατρική απεικόνιση έως καινοτόμες διαμορφώσεις δεδομένων για συστήματα αντίληψης UAV και αρχιτεκτονικές βαθιάς μάθησης για την αναγνώριση εικόνων. Έχει σημειωθεί σημαντική πρόοδος στην αντιμετώπιση των προκλήσεων που τίθενται από τα δεδομένα OOD.

---

Επίσης, αυτή η Διδακτορική Διατριβή υπογραμμίζει την σημασία της ευρωστίας OOD σε κρίσιμες εφαρμογές και υπογραμμίζει την ανάγκη για συνεχή έρευνα και καινοτομία σε αυτόν τον τομέα. Με τη δημιουργία καινοτόμων αλγορίθμων, τη σύνθεση γνώσεων από διάφορες μελέτες και τον εντοπισμό βασικών προκλήσεων και προόδων, αυτή η Διδακτορική Διατριβή στοχεύει να συμβάλει στη συνεχή συζήτηση για την ενίσχυση της αξιοπιστίας και της ασφάλειας των συστημάτων που βασίζονται στην τεχνητή νοημοσύνη σε σενάρια πραγματικού κόσμου. Μέσω διεπιστημονικής συνεργασίας και πειραματισμού, αναπτύξαμε αποτελεσματικές λύσεις που διασφαλίζουν την ανθεκτικότητα και την αποτελεσματικότητα των τεχνολογιών τεχνητής νοημοσύνης σε ιατρικές απεικονίσεις, συστήματα αίσθησης και αποφυγής UAV και τομείς αναγνώρισης εικόνων.

**Λέξεις Κλειδιά:** Ανθεκτικότητα εκτός κατανομής, κρίσιμες εφαρμογές αποστολής, όραση υπολογιστών, βαθιά μάθηση

---

# Ευχαριστίες

Πρώτα από όλα, θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη, για την πολύτιμη καθοδήγηση και υποστήριξη, στον επιβλέποντα καθηγητή μου κ. Στέφανο Κόλλια.

Θα ήθελα επίσης να ευχαριστήσω τον Δρ. Δημήτριο Κόλλια για τη βοήθεια, την καθοδήγηση και τις συμβουλές που μου έδωσε κυρίως στο κρίσιμο αρχικό στάδιο του διδακτορικού.

Θελώ επίσης να εκφράσω τις θερμές μου ευχαριστίες στο Δρ. Χρήστο Σκληρό για την εμπιστοσύνη που μου έδειξε. Η συνεργασία μας έπαιξε καθοριστικό ρόλο τόσο στη διαμόρφωση του διδακτορικού διπλώματος όσο και στην πορεία της μελλοντικής μου έρευνας.

Ακόμα θα ήθελα να ευχαριστήσω τον Ευάγγελο Πετρόγγονα για την άψογη και εποικοδομητική συνεργασία μας.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου Παναγιώτη και Ζωή - Ζωή και Παναγιώτη την αδερφή μου Ελίνα και τον καλύτερο μου φίλο Ορφέα που ήταν πάντα δίπλα μου και με στήριξαν.

---



# Contents

|  |             |
|--|-------------|
| <b>Extended Abstract</b>   | <b>xi</b>   |
| <b>Extended Abstract in Greek</b>                                | <b>xv</b>   |
| <b>Acknowledgments</b>   | <b>xix</b>  |
| <b>List of Figures</b>   | <b>xxv</b>  |
| <b>List of Tables</b>  | <b>xxix</b> |
| <b>1 Introduction to Robustness in Computer Vision</b>           | <b>1</b>    |
| 1.1 Computer Vision: From Perception to Challenges . . . . .     | 2           |
| 1.2 Addressing Out-of-Distribution Robustness . . . . .          | 3           |
| 1.3 Formalization of Out-of-Distribution Robustness . . . . .    | 3           |
| 1.3.1 Multi-Source Domain Generalization . . . . .               | 4           |
| 1.3.2 Single-Source Domain Generalization . . . . .              | 4           |
| 1.4 Types of methods for achieving robustness in computer vision | 5           |
| 1.4.1 Data Manipulation . . . . .                                | 6           |
| 1.4.2 Representation Learning . . . . .                          | 9           |
| 1.4.3 Learning Strategy . . . . .                                | 17          |
| <b>2 Robustness in Image recognition</b>                         | <b>19</b>   |
| 2.1 Introduction . . . . .                                       | 19          |
| 2.2 Materials and Method . . . . .                               | 22          |
| 2.2.1 Transformation Component (TC) . . . . .                    | 22          |
| 2.2.2 Domain Augmentation Generator . . . . .                    | 23          |
| 2.2.3 Learning Objective . . . . .                               | 26          |
| 2.2.4 Overall Objective . . . . .                                | 26          |
| 2.3 Experimental Results . . . . .                               | 27          |

|          |  |           |
|----------|--|-----------|
| 2.4      | Uncertainty Estimation . . . . .                                       | 28        |
| 2.4.1    | Performance Analysis . . . . .   | 29        |
| 2.4.2    | Practical Benefits . . . . .   | 29        |
| 2.4.3    | Robustness and Reliability . . . . .                                   | 29        |
| 2.5      | Conclusion . . . . .   | 32        |
| <b>3</b> | <b>Robustness in Medical Imaging</b>                                   | <b>33</b> |
| 3.1      | Introduction . . . . .   | 33        |
| 3.2      | The COV19-CT-DB Database . . . . .                                     | 35        |
| 3.2.1    | Introduction to the Database . . . . .                                 | 35        |
| 3.2.2    | Data Collection and Processing . . . . .                               | 37        |
| 3.2.3    | Annotation Process . . . . .   | 37        |
| 3.2.4    | Database Composition . . . . .   | 38        |
| 3.2.5    | Ethical Considerations . . . . .                                       | 39        |
| 3.3      | RACNet: The Proposed Architecture . . . . .                            | 43        |
| 3.3.1    | 3-D Analysis Component . . . . .                                       | 44        |
| 3.3.2    | Routing Component . . . . .  | 45        |
| 3.3.3    | Classification Component . . . . .                                     | 47        |
| 3.3.4    | Latent Variable Analysis and Anchor Set Generation . . . . .           | 47        |
| 3.4      | Experimental Study . . . . .   | 48        |
| 3.4.1    | Training RACNet on the COV19-CT-DB Database . . . . .                  | 49        |
| 3.4.2    | Application and Retraining on Additional Databases . . . . .           | 49        |
| 3.5      | RACNet Training: Implementation Details . . . . .                      | 50        |
| 3.5.1    | Architecture Specifications . . . . .                                  | 51        |
| 3.5.2    | Data Preprocessing . . . . .   | 51        |
| 3.5.3    | Training Protocols . . . . .   | 52        |
| 3.5.4    | Hyperparameter Tuning . . . . .  | 53        |
| 3.5.5    | Computational Resources . . . . .                                      | 53        |
| 3.5.6    | Evaluation Metrics . . . . .   | 54        |
| 3.6      | Experimental study . . . . .   | 55        |
| 3.6.1    | Experiments & Ablation Study on COV19-CT-DB . . . . .                  | 55        |
| 3.6.2    | Experiments on COV19D ICCV & ECCV Competitions . . . . .               | 57        |
| 3.6.3    | Experiments on CC-CCII, MosMedData and CT-image<br>Databases . . . . . | 58        |
| 3.6.4    | Anchor Set Creation . . . . .  | 61        |
| 3.6.5    | Anchor Set Unification across Databases . . . . .                      | 66        |

|          |  |            |
|----------|--|------------|
| 3.7      | Deployment . . . . .   | 68         |
| 3.7.1    | MLOps Orchestration . . . . .  | 70         |
| 3.8      | Conclusion and Future Work . . . . .   | 74         |
| <b>4</b> | <b>Robustness in UAV Sense and Avoid</b>   | <b>77</b>  |
| 4.1      | Introduction . . . . .   | 77         |
| 4.2      | NEFELI Architecture . . . . .  | 78         |
| 4.2.1    | Air-to-Air Object Detection . . . . .  | 80         |
| 4.2.2    | Synthetic Common Corruptions for Enhancing Robustness in Air-to-Air Object Detection . . . . .         | 82         |
| 4.2.3    | Aerial Object Tracking . . . . .   | 87         |
| 4.2.4    | Monocular Distance Estimation . . . . .  | 90         |
| 4.3      | Evaluation Methodology and Training Strategy . . . . .   | 94         |
| 4.3.1    | Air-to-Air Object Detection: Benchmark Datasets and Evaluation Metrics . . . . .                       | 95         |
| 4.3.2    | Aerial Object Tracking: Benchmark Dataset, Re-identification Dataset, and Evaluation Metrics . . . . . | 97         |
| 4.3.3    | Monocular Distance Estimation: Experimental Setup  | 101        |
| 4.4      | Benchmark Evaluation . . . . .   | 104        |
| 4.4.1    | Air-to-Air Object Detection . . . . .  | 105        |
| 4.4.2    | Tracking Appearance (Re-identification) . . . . .  | 109        |
| 4.4.3    | Fused Appearance and Motion Tracking . . . . .   | 110        |
| 4.4.4    | Monocular Distance estimation . . . . .  | 113        |
| 4.4.5    | Comparison of NEFELI’s Pipeline with State-of-the-Art Methods . . . . .                                | 118        |
| 4.5      | Discussion of Real-World Experimental Results . . . . .  | 119        |
| 4.5.1    | Edge Implementation . . . . .  | 120        |
| 4.5.2    | Evaluation on Real-World Experiments . . . . .   | 121        |
| 4.5.3    | Enhancing Generalization to Real Flights through Fine-Tuning on Synthetic Corruptions Data . . . . .   | 125        |
| 4.5.4    | Correlation between Robustness to Corruptions and Model Generalization . . . . .                       | 126        |
| 4.6      | Discussion of Limitations and Failure Cases . . . . .  | 126        |
| 4.7      | Summary and Concluding Remarks . . . . .   | 128        |
| <b>5</b> | <b>Conclusion</b>  | <b>131</b> |
| 5.1      | Future Work . . . . .  | 133        |

*Contents*

---

|                       |            |
|-----------------------|------------|
| <b>Greek Glossary</b> | <b>135</b> |
| <b>Publications</b>   | <b>139</b> |
| <b>Bibliography</b>   | <b>143</b> |

# List of Figures

|     |  |    |
|-----|--|----|
| 2.1 | The overall framework of the proposed CUDGNet. The Task Model $M$ and the domain augmentation Generator $G$ are jointly trained, while the transformation component $TC$ and style mixing (EFDMix) further enrich the augmentation capacity. The contrastive loss guides semantically similar samples from different domains to be closer in the embedding space. . . . .  | 21 |
| 2.2 | Estimation of Uncertainty CIFAR-10-C. Our domain uncertainty prediction aligns with Bayesian uncertainty, while our approach is significantly faster. . . . .  | 30 |
| 3.1 | Histogram of CT scan lengths in COV19-CT-DB . . . . .  | 38 |
| 3.2 | Slices from a non COVID-19 CT scan in COV19-CT-DB . . .  | 40 |
| 3.3 | Slices from a COVID-19 CT scan in COV19-CT-DB . . . . .  | 41 |
| 3.4 | Four CT scan slices in COV19-CT-DB, the top 2 from a non-COVID-19 scan and the bottom 2 from a COVID-19, including bilateral ground glass regions in lower lung lobes. . . . .   | 42 |
| 3.5 | The proposed Pipeline: A 3-D input composed of up to $t$ chest CT slices is processed for COVID-19 diagnosis. A CNN-RNN architecture performs 3-D analysis, while a routing mechanism with an 'alignment' step and a Mask Layer handles the varying input length $t$ . A dense layer precedes the output layer that provides the COVID-19 diagnosis; the neuron outputs of the dense layer are further analyzed through clustering to derive a latent variable model and a related set of anchors that provide additional insights into the decision-making process. . . . . | 44 |

|      |  |    |
|------|--|----|
| 3.6  | The Routing Mechanism: (a) without and (b) with the 'alignment' step . . . . .   | 46 |
| 3.7  | Slices from cluster center 0 of COVID-19 category in COV19-CT-DB. Bilateral ground glass regions are seen especially in lower lung lobes. . . . .  | 65 |
| 3.8  | Slices from COVID-19 cluster center 2 in COV19-CT-DB, which is consistent with COVID-19 pneumonia bilateral thickening infiltrates. . . . .  | 65 |
| 3.9  | Slices from non COVID-19 cluster center 9 in COV19-CT-DB   | 66 |
| 3.10 | Slices of a new COVID-19 anchor in CC-CCII database, showing ground glass regions in the lungs. . . . .  | 67 |
| 3.11 | The 5 derived cluster centers in the CT-image Database; top three correspond to non-COVID-19 and bottom two to COVID-19 categories. . . . .  | 68 |
| 3.12 | Architecture diagram illustrating the MLPod™ framework, depicting the flow of information between its constituent modules. Each module serves as an independent service, with rows indicating the direction of data flow. All communications are secured via encryption, highlighting MLPod™ as a comprehensive MLOps sandbox environment. . . . . | 71 |
| 3.13 | Screenshot of the RACNet-based COVID-19 detection application showing the initial data input interface for end-users. .  | 74 |
| 3.14 | Screenshot depicting the execution of the RACNet-based COVID-19 detection application, highlighting the visualization of pipeline steps managed by LogicPod. . . . .   | 75 |
| 3.15 | Illustration of the final diagnostic report generated by the RACNet model within the LogicPod framework. The report provides diagnostic outcomes, textual explanations, and representative medical images, aiding in clinical decision-making.   | 76 |
| 4.1  | Overview of NEFELI's System Pipeline: Depicting the Detection and Tracking Modules. . . . .  | 79 |
| 4.2  | Sliced inference illustrated process (example with 4 slices). . .  | 81 |
| 4.3  | An example of the slices of a high-resolution image. . . . .   | 82 |
| 4.4  | Visualization of the seven corruption types for each severity level in our benchmark . . . . .   | 85 |

|      |  |     |
|------|--|-----|
| 4.5  | Examples of substantial variances in viewpoints and scales of aerial objects from the airborne-Re-ID dataset used to train NEFELI’s appearance model . . . . . | 98  |
| 4.6  | Nefeli’s system pipeline . . . . .   | 101 |
| 4.7  | Collision avoidance/safe separation thresholds . . . . .   | 102 |
| 4.8  | Pipeline of the proposed depth estimation model . . . . .  | 104 |
| 4.9  | Depth estimation visualization for L1, Berhu and multi loss respectively . . . . .   | 117 |
| 4.10 | Depth estimation ground truth and prediction mask for each classification bin . . . . .  | 118 |
| 4.11 | NEFELI Edge Implementation . . . . .   | 120 |
| 4.12 | Overview of the components to implement NEFELI . . . . .   | 122 |
| 4.13 | Detection results in diverse conditions (below the horizon, close distance, intense lighting, left to right) . . . . .   | 123 |





# List of Tables

|     |   |    |
|-----|---|----|
| 2.1 | The accuracy of single-source domain generalisation (%) on CIFAR-10-C. Models are trained on CIFAR-10 and evaluated on the CIFAR-10-C. . . . .  | 27 |
| 2.2 | The accuracy of single-source domain generalisation (%) on PACS. Models are trained on photo and evaluated on the rest of the target domains. . . . .   | 28 |
| 2.3 | Ablation study of the key components of CUDGNet . . . . .   | 30 |
| 3.1 | Attributes of the COV19-CT-DB . . . . .   | 39 |
| 3.2 | Performance comparison between RACNet and other state-of-the-art networks on the test set of COV19-CT-DB database (non-segmented data) . . . . .  | 55 |
| 3.3 | Ablation Study: Performance comparison on the test set of COV19-CT-DB database (non-segmented data) . . . . .   | 56 |
| 3.4 | Performance comparison between RACNet and the state-of-the-art on the test set of COV19D-ICCV2021 and COV19D-ECCV2022 databases of the respective ICCV and ECCV Competitions; F1 Score presented in % . . . . . | 58 |
| 3.5 | Performance comparison between RACNet and the state-of-the-art on the test set of CC-CCII, MosMedData, and CT-image Database; Acc stands for Accuracy metric . . . . .  | 61 |
| 3.6 | Number of CT Scans per cluster, cluster category &Severity category in COV19-CT-DB . . . . .  | 63 |
| 3.7 | Description of the Severity Categories . . . . .  | 63 |
| 3.8 | Description of findings in each cluster center in COV19-CT-DB   | 64 |

|     |  |     |
|-----|--|-----|
| 4.1 | The benchmarking results of 8 object detectors on AOT and AOT-C in terms of Average Precision (AP), inference speed (fps), and model size (M) . . . . .  | 107 |
| 4.2 | The benchmarking results of 8 object detectors on AOT-C. We show the performance under each corruption and the overall corruption robustness $AP_{cor}$ averaged over all corruption types . . . . .   | 108 |
| 4.3 | Results of state-of-the-art methods on our aerial object re-id dataset named airborne-ReID. . . . .  | 110 |
| 4.4 | Comparison of proposed detection and tracking methods on AOT dataset. Our fused detection (Enhanced YOLOv5) and tracking method with appearance model (OSNet) trained on our airborne-ReID dataset has the best result in terms of the primary tracking metrics. . . . . | 112 |
| 4.5 | Regression metrics for the different losses . . . . .  | 115 |
| 4.6 | Classification metrics for the different losses . . . . .  | 118 |
| 4.7 | Ablation study on NEFELI’s key components using real-flight data. Each component significantly improves performance, demonstrating NEFELI’s robustness on data differing from the training set. . . . .  | 124 |
| 4.8 | Evaluation results (measured in Average Precision (AP)) on the AOT test set and our real-world flight tests using corruptions as augmentations (Finetuned) and without (Base). . . . .   | 125 |

# Chapter 1

## Introduction to Robustness in Computer Vision

Humans perceive their surroundings through various senses, including touch, taste, sound, smell, and sight, each contributing to our understanding of the world. Visual perception holds particular significance, aiding in our comprehension from early childhood through observation of human interactions, object movements, and light reflections. Over time, we develop the ability to scan our surroundings for visual cues that inform us about potential object trajectories, safe pathways, and potential obstacles. Visual media, encompassing photographs, videos, artwork, and written content, plays a crucial role in conveying information, providing entertainment, and fostering cross-cultural understanding.

In recent years, modern machine learning techniques have demonstrated impressive capabilities across diverse fields such as natural language processing, computer vision, and recommendation systems. While excelling in controlled settings, these methods have shown vulnerability to shifts in data distribution, posing significant challenges in critical domains like healthcare and autonomous driving. Even minor inaccuracies in such applications can lead to catastrophic outcomes. Consequently, there is growing interest in exploring Out-of-Distribution (OOD) generalization to enhance the robustness and reliability of intelligent systems across real-world scenarios.

## **1.1 Computer Vision: From Perception to Challenges**

Can machines meaningfully grasp the visual world? Computer vision endeavors to construct models capable of understanding visual data. Approximately a decade ago, the advent of deep learning revolutionized computer vision with the breakthrough AlexNet model [1], which significantly reduced error rates in challenging image recognition tasks. Since then, deep learning has become the dominant approach, widely adopted in consumer products, autonomous driving, medical diagnostics, and more.

Training deep neural networks involves optimizing numerous parameters defining artificial neurons, demanding vast amounts of data for effective calibration. Traditionally, these models have excelled on meticulously curated datasets like ImageNet-1K, surpassing human-level performance in many cases. However, they exhibit fragility to imperceptible adversarial noise, highlighting their sensitivity to controlled lab conditions.

Beyond adversarial examples, deep learning models struggle with generalization across diverse real-world data. Challenges arise from distribution shifts between training and test environments, where differences in image features affect model performance. For instance, a classifier trained on specific cat and dog colors may falter when confronted with variations in real-world scenarios.

As deep learning permeates everyday life and interdisciplinary research, understanding its behavior across real-world settings is crucial. Failures can have profound implications for safety and societal equity, necessitating robust perception systems resilient to distribution shifts. This dissertation explores these challenges comprehensively.

## 1.2 Addressing Out-of-Distribution Robustness

Addressing Out-of-Distribution (OOD) robustness entails several critical avenues. Firstly, formal characterization of distribution shifts between training and test data is essential, given their disparate origins. Currently, consensus is lacking in OOD generalization literature, with varied approaches to modeling potential test distributions. Causal learning techniques attribute shifts to causal structures, while invariant learning focuses on real-world scenarios. Stable learning methods introduce shifts through selection bias.

Secondly, developing algorithms with robust OOD generalization performance is pivotal. Methodological branches include unsupervised representation learning, supervised model learning, and optimization techniques. Each approach aims to enhance model resilience against distribution shifts.

Thirdly, evaluating OOD generalization performance presents challenges, requiring curated datasets and metrics beyond traditional benchmarks. Effective evaluation frameworks are essential to gauge model efficacy in diverse real-world contexts.

## 1.3 Formalization of Out-of-Distribution Robustness

Domain Generalization (DG) and Out-of-Distribution (OOD) robustness are fundamental concepts in computer vision, aiming to enhance a model's ability to generalize beyond its training data. DG focuses on training models to perform well on unseen domains by learning invariant features from multiple source domains. In contrast, OOD robustness specifically addresses a model's resilience to unexpected variations and corruptions in input data. Despite their distinct focuses, both concepts aim to achieve robust and reliable performance in real-world scenarios.

Among DG approaches, Single-Source Domain Generalization (SSDG) stands

out as particularly practical for mission-critical applications. SSDG assumes training data from a single domain, which is more applicable when collecting diverse multi-source data is impractical. Therefore, developing models that can generalize effectively from a single domain is crucial for ensuring reliability and safety in applications like autonomous driving, medical imaging, and aerial surveillance.

Let  $X$  denote the input (feature) space and  $Y$  the target (label) space. A domain is defined as a joint distribution  $P_{XY}$  on  $X \times Y$ . For a specific domain  $P_{XY}$ ,  $P_X$  represents the marginal distribution on  $X$ ,  $P_{Y|X}$  denotes the posterior distribution of  $Y$  given  $X$ , and  $P_{X|Y}$  refers to the class-conditional distribution of  $X$  given  $Y$ .

In DG,  $K$  similar but distinct source domains  $S = \{S_k = \{(x^{(k)}, y^{(k)})\}_{k=1}^K\}$  are available, each associated with a joint distribution  $P_{XY}^{(k)}$ . The objective is to train a predictive model  $f : X \rightarrow Y$  using these source domain data such that the prediction error on an unseen target domain  $T = \{x_T\}$  is minimized.

### **1.3.1 Multi-Source Domain Generalization**

Multi-Source DG utilizes multiple distinct source domains (i.e.,  $K > 1$ ) to learn representations that are invariant to different marginal distributions. This approach leverages the diversity across source domains to discover stable patterns that generalize effectively to unseen domains.

### **1.3.2 Single-Source Domain Generalization**

In contrast, Single-Source DG assumes homogeneous training data sampled from a single domain ( $K = 1$ ). This setting is closely related to OOD robustness, which investigates model resilience under unexpected variations and corruptions in input data. Single-Source DG methods do not require domain labels for learning, making them versatile across both single- and

multi-source scenarios. Many existing methods addressing Single-Source DG provide generic solutions for OOD generalization, demonstrating efficacy across diverse datasets.

## 1.4 Types of methods for achieving robustness in computer vision

Since its formal inception in 2011 by Blanchard et al. [2], numerous methods have emerged to tackle the challenge of out-of-distribution (OOD) generalization [3–9]. These approaches encompass diverse strategies, including aligning source domain distributions for domain-invariant representation learning [10, 11], exposing models to domain shift through meta-learning [12, 13], and augmenting data using domain synthesis techniques [14, 15], among others. Beyond computer vision, domain generalization (DG) has been extensively explored in applications such as object recognition [16, 17], semantic segmentation [18, 19], person re-identification [19, 20], speech recognition [21], natural language processing [13], medical imaging [22, 23], and reinforcement learning [20]. This chapter provides a comprehensive literature review on OOD robustness in computer vision, focusing on learning algorithms developed over the past decade and outlining future research directions.

In this section, we categorize existing OOD robustness methods into three groups:

- **Data Manipulation:** This group focuses on modifying input data to facilitate learning general representations. Techniques include data augmentation, which involves randomizing and transforming input data, and data generation, which creates diverse samples to enhance generalization.
- **Representation Learning:** Widely used in DG, these methods include domain-invariant representation learning (e.g., adversarial train-

ing, feature alignment) and feature disentanglement to improve generalization.

- **Learning Strategy:** These methods leverage general learning strategies such as ensemble learning, meta-learning, gradient manipulation, distributionally robust optimization, and self-supervised learning to enhance overall generalization capacity.

These categories are distinct yet complementary, often combined to achieve enhanced performance. Detailed descriptions of each approach within these categories are provided subsequently.

### 1.4.1 Data Manipulation

In machine learning, the quest for more training data is ongoing as model generalization heavily relies on both volume and diversity. Data manipulation in DG serves to augment existing datasets, thereby enhancing model generalization capabilities. Methods include data augmentation to diversify training data and data generation to synthesize additional samples. The overarching goal is to minimize the expected loss over both original and manipulated data:

$$\min \mathbb{E}_{x,y}[l(h(x), y)] + \mathbb{E}_{x',y}[l(h(x'), y)]$$

where  $x' = M(x)$  denotes manipulated data obtained using function  $M(\cdot)$ .

#### Data Augmentation for Domain Generalization

Data augmentation is a highly effective method in training machine learning models. Common augmentation techniques include flipping, rotation, scaling, cropping, adding noise, and others. These methods are widely used in supervised learning to enhance model generalization by mitigating overfitting [24, 25]. Similarly, they are applicable in domain generalization



(DG), where the function  $M(\cdot)$  can utilize these data augmentation functions.

### **Domain Randomization**

In addition to conventional augmentation methods, domain randomization is a powerful technique for data augmentation. It involves generating new data to simulate complex environments based on limited training samples. In this context, the function  $M(\cdot)$  applies various manual transformations, particularly effective in image data. These transformations include altering object location and texture, changing object number and shape, adjusting illumination and camera view, and introducing diverse types of random noise. Tobin et al. [26] pioneered this approach, generating additional training data from simulated environments to enhance generalization in real-world scenarios. Similar strategies have been employed in [27–29] to improve model generalization. Prakash et al. [30] extended this idea by considering scene structure when randomly placing objects for data generation, enabling the neural network to leverage context in object detection. Moreover, [31] proposed augmenting not only features but also labels. While randomization enhances sample diversity, it is crucial to refine irrelevant randomizations to optimize model efficiency.

### **Adversarial Data Augmentation**

Adversarial data augmentation aims to optimize generalization by diversifying data while preserving reliability. Shankar et al. [32] employed a Bayesian network to model the relationship between label, domain, and input instance, introducing CrossGrad, a cautious data augmentation approach that perturbs input along significant domain change directions while minimizing alterations to class labels. Volpi et al. [33] developed an iterative method augmenting the source dataset with examples from a hypothetical target domain challenging the current model, appending adversarial examples to facilitate adaptive data augmentation at each iteration. Zhou et al. [15] utilized adversarial training of a transformation network for data augmentation, diverging from direct input modification through gradient ascent while integrating weak and strong augmentation regularization as

in [34]. Adversarial data augmentation often involves specific optimization objectives exploitable by networks, though its optimization typically involves adversarial training, posing challenges.

## **Data Generation**

Data generation is a popular technique for enriching data diversity to improve model generalization. Here, function  $M(\cdot)$  can leverage various generative models like Variational Autoencoder (VAE) [35], Generative Adversarial Networks (GAN) [36], and the Mixup [37] strategy.

Rahman et al. [38] applied ComboGAN [37] to generate new data and utilized domain discrepancy measures such as Maximum Mean Discrepancy (MMD) [39] to minimize distribution divergence between real and generated images, aiding in learning general representations. Qiao et al. [?] employed adversarial training to create challenging yet fictitious populations, using a Wasserstein Auto-Encoder (WAE) [40] for generating samples preserving semantic content and exhibiting substantial domain transportation. [41] introduced novel distributions under semantic consistency, optimizing the difference between source and novel distributions. Somavarapu et al. [42] proposed simple image stylization-based transformations to explore cross-source variability for enhanced generalization, using Adaptive Instance Normalization (AdaIN) [40] for rapid stylization to diverse styles. Differing from others, [38] utilized adversarial training to generate domains rather than individual samples, adding complexity through diverse generative models, necessitating attention to model capacity and computational overhead.

Additionally, Mixup [37] is another popular technique for data generation. Mixup creates new data by linearly interpolating between any two instances and their labels, using weights sampled from a Beta distribution, avoiding the need for training generative models. Recent methods employing Mixup for Domain Generalization (DG) perform Mixup either in the original space [43] to generate new samples or in the feature space [44], avoiding explicit generation of raw training samples. These approaches show promising performance on prominent benchmarks while maintaining conceptual

and computational simplicity.

### 1.4.2 Representation Learning

Representation learning has been a central focus in machine learning for a considerable period [45], contributing significantly to the success of domain generalization. The prediction function  $h$  is often decomposed into two components:  $h = f \circ g$ , where  $g$  denotes the representation learning function and  $f$  represents the classifier function. The goal of representation learning is typically expressed as:

$$\min_{f,g} \mathbb{E}_{x,y}[l(f(g(x)), y)] + \lambda l_{\text{reg}}$$

where  $l_{\text{reg}}$  is a regularization term and  $\lambda$  is a tradeoff parameter. Various techniques have been developed to enhance learning of the feature extraction function  $g$  alongside the corresponding regularization  $l_{\text{reg}}$ .

#### Domain-Invariant Representation-Based DG

The study by [45] provided theoretical evidence that maintaining invariant feature representations across different domains enhances generalizability and transferability. Numerous algorithms have since been developed for domain adaptation and domain generalization, aiming to minimize representation gaps among multiple source domains within a specific feature space to achieve domain invariance.

#### Kernel-Based Methods

Kernel-based methods represent a classical learning paradigm in machine learning. These methods employ kernel functions to map original data into a high-dimensional feature space without explicitly computing the coordinates in that space. Instead, they rely on computing inner products between samples in the feature space. Support Vector Machines (SVM) [46] are one

of the most well-known kernel-based methods. In the context of domain generalization, several algorithms based on kernel methods have been developed, where the representation learning function  $g$  utilizes a feature map  $\phi(\cdot)$  computed via kernel functions like the Radial Basis Function (RBF) kernel and the Laplacian kernel.

[47] were among the pioneers applying kernel methods to domain generalization, further expanded upon in [31]. They employed positive semi-definite kernel learning to derive a domain-invariant kernel from training data. Grubinger et al. [48] adapted Transfer Component Analysis (TCA) [46] to minimize multi-domain distance for domain generalization. Domain-Invariant Component Analysis (DICA) [49], similar to TCA, utilizes kernels for domain generalization by finding a feature transformation kernel  $k(\cdot, \cdot)$  that reduces distribution discrepancies among all data points in the feature space. Gan et al. [47] extended DICA with attribute regularization. Conversely, Li et al. [50] focused on learning feature representations with domain-invariant class-conditional distributions. Scatter Component Analysis (SCA) [51] applied Fisher’s discriminant analysis to minimize representation discrepancies within the same class and domain while maximizing discrepancies across different classes and domains. Erfani et al. [52] introduced Elliptical Summary Randomization (ESRand), using randomized kernel and elliptical data summarization. ESRand projects each domain onto an ellipse to represent domain information and computes distances using a similarity metric. Hu et al. [53] proposed multi-domain discriminant analysis, employing class-wise kernel learning for domain generalization, providing a more granular approach. Overall, these methods within this category often intertwine with other approaches, serving as divergence measures or theoretical foundations.

## **Domain Adversarial Learning**

Domain-adversarial training is a widely adopted approach for learning features that are invariant across domains. Ganin and Lempitsky [54] and Ganin et al. [42] introduced the Domain-adversarial Neural Network (DANN) for domain adaptation, where both the generator and discriminator are adversarially trained. The discriminator’s goal is to differentiate between do-

mains, while the generator aims to deceive the discriminator by learning domain-invariant feature representations. Li et al. [55] extended this concept to domain generalization. Gong et al. [56] utilized adversarial training to progressively reduce domain discrepancy in a manifold space. Li et al. [57] proposed the Conditional Invariant Adversarial Network (CIAN), employing class-wise adversarial networks for domain generalization. Similar strategies were explored in [42]. Jia et al. [58] employed single-side adversarial learning and asymmetric triplet loss to ensure that only real faces from different domains were indistinguishable, thereby improving generalization of class boundaries for unseen domains. Furthermore, Zhao et al. [59] introduced entropy regularization by minimizing the Kullback-Leibler (KL) divergence between conditional distributions of different training domains to encourage the network to learn domain-invariant features. Several other Generative Adversarial Network (GAN)-based methods [58] have been proposed, with theoretically guaranteed generalization bounds.

### **Explicit Feature Alignment**

This line of research focuses on aligning features across source domains to learn domain-invariant representations through explicit feature distribution alignment [60] or feature normalization [58]. Motiian et al. [43] introduced a cross-domain contrastive loss for representation learning, ensuring that mapped domains are semantically aligned yet maximally separated. Some methods explicitly minimize feature distribution divergence by minimizing metrics like Maximum Mean Discrepancy (MMD) [59], second-order correlation, mean and variance (moment matching) [61], and Wasserstein distance [61] between domains, applicable to both domain adaptation and domain generalization. Zhou et al. [62] aligned the marginal distribution of different source domains via optimal transport, minimizing Wasserstein distance to achieve a domain-invariant feature space.

Additionally, some works leverage feature normalization techniques to enhance domain generalization capability [58]. Pan et al. [63] introduced Instance Normalization (IN) layers to CNNs, improving model generalization by eliminating instance-specific style discrepancies. IN has been extensively studied in image style transfer [64], where image style is reflected by IN pa-

rameters, i.e., mean and variance of each feature channel. IN layers [58] can remove instance-specific style discrepancies but may inadvertently remove discriminative information. In IBNNet, IN and Batch Normalization (BN) are used in parallel to preserve some discriminative information. In [63], BN layers are replaced by Batch-Instance Normalization (BIN) layers, adaptively balancing BN and IN for each channel by selectively using BN and IN. Jin et al. [65,66] proposed a Style Normalization and Restoration (SNR) module to simultaneously ensure high generalization and discrimination capability of networks. Following style normalization by IN, a restitution step distills task-relevant discriminative features from residuals (i.e., the difference between original and style-normalized features), reintegrating them into the network to ensure high discrimination. This restitution concept has been extended to other alignment-based methods to restore helpful discriminative information that may have been lost during alignment [67]. Recently, Qi et al. [68] applied IN to unsupervised domain generalization, where no labels are available in training domains, to acquire invariant and transferable features. A combination of different normalization techniques is presented in [69], demonstrating that adaptively learning normalization techniques can enhance domain generalization. This category of methods is flexible and applicable across other approaches.

### **Invariant Risk Minimization (IRM)**

Arjovsky et al. [70] proposed an alternative perspective on domain-invariant representation for domain generalization. Instead of matching representation distributions across all domains, they focused on ensuring that the optimal classifier on top of the representation space remains consistent across all domains. The underlying idea is that the ideal representation for prediction should be influenced solely by the target variable  $y$  and remain invariant to other factors, thus ensuring domain-invariant representations. However, solving this problem is challenging due to an inner-level optimization problem within its constraints. Hence, the authors introduced a surrogate problem that facilitates learning of the feature extractor  $g$  by considering a dummy representation-level classifier  $f = 1$  and using a gradient norm term to measure the optimality of this classifier. Additionally, the work provides a generalization theory under a potentially strong linear assumption, sug-

gesting that with a sufficient number of source domains, the ground-truth invariant classifier can be identified.

### **Invariant Risk Minimization (IRM)**

IRM has gained significant attention recently, with further theoretical analyses on its efficacy [71] and instances where it fails [72]. Moreover, IRM has been extended to other domains such as text classification [73] and reinforcement learning [74]. The concept of ensuring invariance of the optimal representation-level classifier has also been expanded upon. Krueger et al. [70] promote this invariance by minimizing the extrapolated risk across source domains, effectively reducing the variance in risks associated with different source domains. Mitrovic et al. [31] aim to learn such representations in a self-supervised setup, where a second domain is created through data augmentation, introducing various semantically irrelevant variations. Recently, [75] discovered that ensuring invariance of  $f$  alone is inadequate. They found that IRM can still fail if  $g$  captures "fully informative invariant features," resulting in  $y$  being independent of  $x$  across all domains. This finding is particularly pertinent in classification tasks compared to regression tasks. To address this, they introduce an information bottleneck regularization to retain only partially informative features.

### **Feature Disentanglement-based Domain Generalization**

Disentangled representation learning aims to develop a function that transforms a sample into a feature vector, encapsulating information about various factors of variation, where each dimension or subset thereof contains information related to specific factors. Disentanglement-based approaches within domain generalization typically decompose a feature representation into interpretable components or sub-features. One component encapsulates domain-shared or invariant information, while others pertain to domain-specific attributes. The optimization objective for disentanglement-based domain generalization can be succinctly summarized as:

$$\min_{g_c, g_s, f} \mathbb{E}_{x, y} [l(f(g_c(x)), y)] + \lambda l_{\text{reg}} + \mu l_{\text{recon}}([g_c(x), g_s(x)], x)$$

Here,  $g_c$  and  $g_s$  represent domain-shared and domain-specific feature representations, respectively. Parameters  $\lambda$  and  $\mu$  are trade-off coefficients. The regularization term  $l_{\text{reg}}$  explicitly encourages segregation of domain-shared and specific features, while  $l_{\text{recon}}$  denotes a reconstruction loss aimed at preventing information loss. It's important to note that  $[g_c(x), g_s(x)]$  represents a fusion of two types of features, not merely a concatenation operation. Depending on network architecture and implementation mechanisms, disentanglement-based domain generalization can primarily be categorized into three types: multi-component analysis, generative modeling, and causality-inspired methods.

### Multi-component Analysis

In multi-component analysis, domain-shared and domain-specific features are typically extracted using parameters from domain-shared and domain-specific networks. The UndoBias method [76] originated with an SVM model aimed at maximizing interval classification on all training data for domain generalization. They represent the parameters of the  $i$ -th domain as  $w_i = w_0 + \Delta_i$ , where  $w_0$  represents domain-shared parameters and  $\Delta_i$  represents domain-specific parameters. Other methods have extended the UndoBias concept from various perspectives. Niu et al. [75] propose using multi-view learning for domain generalization, introducing Multi-view DG (MVDG) to learn combinations of exemplar SVMs under different views for robust generalization. Ding and Fu [77] design domain-specific networks for each domain and a shared domain-invariant network for all domains to learn disentangled representations, using low-rank reconstruction to align these two networks in a structured manner. Li et al. [72] adapt the UndoBias idea to the neural network context, developing a low-rank parameterized CNN model for end-to-end training. Zunino et al. [78] learn disentangled representations by manually comparing attention heat maps from different domains. Various other works also adopt multi-component analysis for disentanglement [77]. In essence, multi-component analysis can be implemented across



different architectures and remains effective for representation disentanglement.

### Generative Modeling

Generative models offer another perspective on disentanglement by focusing on the process of data generation. These methods seek to understand the generative mechanisms of samples at domain, sample, and label levels. Some approaches further disentangle inputs into class-irrelevant features containing information specific to individual instances [78]. The Domain-invariant variational autoencoder (DIVA) [79] disentangles features into domain information, category information, and other information within the VAE framework. Peng et al. [80] separate fine-grained domain information and category information learned within VAEs. Qiao et al. [81] also leverage VAE for disentanglement, proposing a Unified Feature Disentanglement Network (UFDN) treating both data domains and image attributes of interest as latent factors to disentangle. Similarly, Zhang et al. [82] isolate semantic and variational parts of samples. Comparable approaches include [46]. [82] suggest disentangling style and other information using generative models, their method serving both domain adaptation and domain generalization. Generative models not only enhance OOD performance but also facilitate generation tasks, promising utility across various applications.

Causality provides a deeper understanding of relationships between variables beyond statistical correlations (joint distribution). It offers insights into how systems respond to interventions, making it particularly relevant for transfer learning tasks where domain shifts can be viewed as interventions. Within a causal framework, the ideal representation corresponds to the true cause of the label (e.g., object shape), ensuring predictions remain robust against interventions on correlated yet semantically irrelevant features (e.g., background, color, style). Several studies [83] have explored causality in the context of domain adaptation.

For domain generalization, He et al. [84] reweighted input samples to align the weighted correlation with causal effects. Zhang et al. [85] treated Fourier

features as causal factors in images, enforcing independence among these features. When additional object identity data was available, [86] enforced conditional independence of representations from the domain index given the same object. In cases where object labels were unavailable, [85] learned an object-related feature based on separate stage labels. For single-source domain generalization, [83] used data augmentation to represent information about causal factors. Augmentation operations were designed to simulate outcomes under interventions on irrelevant features, guided by specific domain knowledge.

Generative methods also incorporate causality considerations. Zhang et al. [87] explicitly modeled a manipulation variable causing domain shifts, even when unobserved. Liu et al. [74] leveraged causal invariance for single-source generalization, emphasizing the process’s ability to maintain invariance in generating  $(x, y)$  data based on factors, extending beyond mere inference invariance. They demonstrated the identifiability of causal factors and their beneficial impact on generalization. [71] extended this approach and theory to multiple source domains, where irrelevant factors could be identified with more informative data.

Meta-learning involves acquiring a general model from multiple tasks using optimization-based methods [88], metric-based learning [87], or model-based approaches. This concept has been adapted for domain generalization (DG), where data from various source domains is divided into meta-train and meta-test sets to simulate domain shifts. Let  $\theta$  denote the model parameters to be learned, and meta-learning can be formalized as:

$$\theta^* = \text{Learn}(S_{\text{mte}}; \phi^*) = \text{Learn}(S_{\text{mte}}; \text{MetaLearn}(S_{\text{mtrn}})),$$

where  $\phi^* = \text{MetaLearn}(S_{\text{mtrn}})$  represents the meta-learned parameters from the meta-train set  $S_{\text{mtrn}}$ , used to learn the model parameters  $\theta^*$  on the meta-test set  $S_{\text{mte}}$ . The functions  $\text{Learn}(\hat{u})$  and  $\text{MetaLearn}(\hat{u})$  are implemented by various meta-learning algorithms, addressing a bi-level optimization problem. The gradient update can be expressed as:

$$\theta = \theta - \alpha \frac{\partial}{\partial \theta} (l(S_{\text{mte}}; \theta) + \beta l(S_{\text{mtrn}}; \phi)),$$

where  $\alpha$  and  $\beta$  denote learning rates for the outer and inner loops, respectively.

Finn et al. [88] introduced Model-agnostic meta-learning (MAML). Li et al. [87] extended MAML to MLDG (meta-learning for domain generalization), adapting meta-learning for DG by partitioning source domain data into meta-train and meta-test sets to learn generalized representations. Zhou et al. [89] proposed MetaReg, learning a meta regularizer for classifiers. Wang et al. [41] introduced feature-critic training for the feature extractor using a meta optimizer, while Dou et al. [90] incorporated complementary losses into MLDG.

### 1.4.3 Learning Strategy

#### Gradient operation-based DG

Apart from meta-learning and ensemble learning, recent studies explore gradient-based methods to enforce network learning of generalized representations. Huang et al. [91] introduced self-challenging training, iteratively discarding dominant features and forcing the network to activate remaining features correlating with labels, enhancing generalization. Shi et al. [92] proposed a gradient-matching scheme, aligning gradient directions between domains by maximizing the gradient inner product (GIP):

$$L = L_{cls}(S_{train}; \theta) - \frac{\lambda}{2} \sum_i \sum_j G_i \cdot G_j,$$

where  $G_i$  and  $G_j$  are gradients calculated as  $G = \mathbb{E} \frac{\partial l(x,y;\theta)}{\partial \theta}$ . Other enhancements include adding CORAL [93] loss for gradient invariance [93]

and maximizing neuron coverage with gradient similarity regularization [94]. Wang et al. [80] proposed knowledge distillation based on gradient filtering.

### **Distributionally robust optimization-based DG**

Distributionally robust optimization (DRO) [70] aims to learn models resilient to worst-case distribution scenarios, aligning with DG goals. Sagawa et al. [71] introduced GroupDRO, requiring explicit group annotations initially and later refined to a fraction of the validation set [91]. Other works reduce variance using VRex [64] or class-conditioned Wasserstein DRO [46]. Koh et al. [61] addressed subpopulation shifts with DRO, while Du et al. [95] proposed AdaRNN for DG without explicit group annotations, optimizing worst-case scenarios through an optimization framework.

### **Self-supervised learning-based DG**

Self-supervised learning (SSL) leverages large-scale unlabeled data for self-supervised tasks [43]. Wang et al. [63] pioneered jigsaw puzzles as a self-supervision task for generalized representations. Contrastive learning has gained popularity, contrasting positive and negative pairs [96], applicable in unsupervised DG scenarios lacking labeled domains [68]. SSL also supports pretraining on multi-domain data, facilitating robust model training under domain shifts, albeit with increased computational demands.

### **Other learning strategy for DG**

Various alternative strategies enhance DG. Metric learning [63] refines pairwise distances, while Wang et al. [38] integrate random forests to boost CNN generalization, sampling triplets via forest split probabilities for CNN parameter updates. Model calibration [51] aligns with OOD performance, Wang et al. [95] explore network substructures, and Wang et al. [96] emphasize shape-invariant features. Wang et al. [59] introduce stochastic weight averaging to identify flat minima. As DG evolves, diverse strategies are anticipated to enrich its methodologies.

# Chapter 2

## Robustness in Image recognition

### 2.1 Introduction

In the realm of single domain generalisation, the principal goal is to develop models that, despite being trained exclusively on data from a single domain, can exhibit strong performance when exposed to various unseen domains. This paper presents a novel model, the Contrastive Uncertainty Domain Generalisation Network (CUDGNet), specifically designed to address the challenges of single domain generalisation in image recognition.

The central innovation of CUDGNet is its method of augmenting the source capacity in both the input and label spaces through a fictitious domain generator. This generator operates in conjunction with a contrastive learning framework to simultaneously learn domain-invariant representations for each class. Our approach aims to achieve significant domain expansion from the generator subnetwork while preventing representation collapse, thereby ensuring robust generalisation.

Extensive experiments conducted on two Single Source Domain Generalisation (SSDG) datasets underscore the efficacy of our approach, which outperforms state-of-the-art single-DG methods by up to 7.08%. Additionally, our method facilitates efficient uncertainty estimation during inference through a single forward pass of the generator subnetwork.

The idea of leveraging diversity for model training has been extensively explored. Prior research [97–99] has demonstrated that employing a wide range of augmentations during training significantly enhances a model’s resilience to distribution shifts. When the nature of diversity encountered during testing can be identified, specific augmentations can be applied to mitigate its effects.

Beyond input diversity, SSDG approaches must also focus on learning domain-invariant representations. Numerous previous works [97, 98] have successfully incorporated contrastive learning to achieve this, ensuring that each class forms distinct clusters in the representation space. This clustering facilitates the learning of improved decision boundaries, which are crucial for enhanced generalisation capabilities.

However, previous research [100–102] has often overlooked the risks associated with utilising augmented data for out-of-domain generalisation. This oversight raises significant safety and security concerns, particularly in mission-critical applications. For instance, deploying self-driving vehicles in unfamiliar environments necessitates a comprehensive understanding of predictive uncertainty for effective risk evaluation.

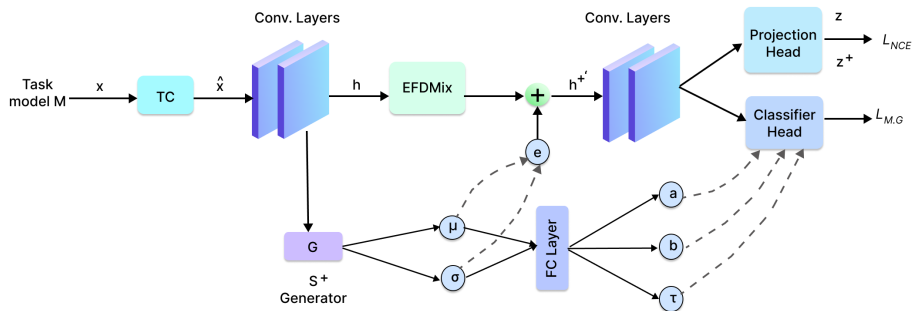
Recently, [103] introduced a Bayesian meta-learning framework that leverages the uncertainty of domain augmentations to improve domain generalisation through a curriculum learning scheme, offering rapid uncertainty assessment. Despite its innovative approach, this framework has limitations, including sensitivity to hyperparameters that can destabilise training and high computational demands that hinder scalability to complex networks.

Drawing inspiration from [103], our approach aims to leverage the uncertainty of domain augmentations in both input and label spaces. To address the limitations of previous works, we propose a novel framework comprising a task model  $M$  and a domain augmentation generator  $G$ . These components enhance each other through collaborative learning. The domain augmentation generator  $G$  produces secure and efficient domains, guided by uncertainty assessment, which are systematically extended to enhance

coverage and comprehensiveness. To ensure cross-domain invariant representation across all generated domains, contrastive learning is integrated into the task model  $M$ .

The main contributions of this paper can be summarised as follows:

- We propose a novel framework that leverages adversarial data augmentation and style transfer for domain expansion while preserving semantic information through contrastive learning.
- Our framework can estimate uncertainty in a single forward pass while achieving state-of-the-art accuracy.
- We validate our framework’s performance through comparative analysis and ablation studies on two SSDG datasets.



**Figure 2.1:** The overall framework of the proposed CUDGNet. The Task Model  $M$  and the domain augmentation Generator  $G$  are jointly trained, while the transformation component  $TC$  and style mixing (EFDMix) further enrich the augmentation capacity. The contrastive loss guides semantically similar samples from different domains to be closer in the embedding space.

## 2.2 Materials and Method

In this section, we provide a detailed outline of the Contrastive Uncertainty Domain Generalisation Network (CUDGNet), as illustrated in Figure 2.1. Our goal is to train a robust model using data from a single domain  $S$ , with the expectation that this model will perform effectively across multiple unseen domain distributions  $\{T_1, T_2, \dots\} \sim p(T)$ . To achieve this, we generate a series of domain augmentations  $\{S_1^+, S_2^+, \dots\} \sim p(S^+)$ , which approximate  $p(T)$ , allowing the task model to learn to generalise to previously unseen domains. Additionally, we demonstrate how to assess the uncertainty of new domains as a byproduct of the perturbations used for adversarial domain augmentation.

To create new domains while preserving class-specific details, we introduce two auxiliary components: the Transformation Component (TC) and the domain augmentation generator  $G$ . The latter is a novel feature perturbation subnetwork that combines style transfer and variational feature perturbations, following a learnable multivariate Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ , for diverse and content-preserved feature augmentations. Furthermore, the domain augmentation learning process is enhanced by incorporating image-structure information generated through our TC, which uses affine transformations and fractals to enrich the input space. To organise domain alignment and classification effectively, we employ contrastive learning to acquire representations that are invariant to domain shifts and to prevent representation collapse due to extreme domain shifts in feature perturbations. This approach facilitates the progressive formation of domain augmentations and well-defined clusters for each class in the representation space.

### 2.2.1 Transformation Component (TC)

The TC transforms the initial image  $x$  from the original domain  $\mathcal{S}$  into a novel image  $\hat{x}$  within the same domain using the following process:



$$\hat{x} = TC(x) = \begin{cases} (x \oplus f_1) \\ (x \oplus f_1) \otimes (x \oplus x_{aff}) \\ (x \oplus f_1) \otimes (x \oplus x_{aff}) \otimes (x \oplus f_2) \end{cases} \quad (2.1)$$

where each of the three branches has an equal probability of occurring.  $x_{aff}$  is the image resulting from an affine transformation (e.g., rotation, translation, contrast adjustment) applied to the initial image  $x$ .  $f_1$  and  $f_2$  denote fractal images [104];  $\oplus$  and  $\otimes$  are element-wise additions and multiplications, respectively.

While the TC may not expand the domain space significantly, it plays a crucial role in our method by helping to avoid representational collapse during the initial epochs of domain expansion when  $G$  may introduce severe noise. The unique structural characteristics of fractals, which are non-random and unlikely to arise from processes of maximum entropy or Gaussian noise, make this transformation orthogonal to the domain augmentation generator  $G$ . In Equation 2.1, the image undergoes  $k$  transformations (with  $k \in [0, 10]$  being a hyperparameter).

### 2.2.2 Domain Augmentation Generator

We illustrate the process of generating the unseen domain  $S^+$  from  $S$  through the generator  $G$ , ensuring that the samples generated adhere to the criteria of safety and effectiveness. Safety denotes that the generated samples preserve semantic information, while effectiveness implies that the generated samples encompass a diverse range of unseen domain-specific details.

**Style manipulation.** In conjunction with the TC, we enrich the input space using style manipulation. We integrate Exact Histogram Matching (EHM) [105] to represent style information by using high-order feature statistics. We use the Sort-Matching algorithm [106] due to its efficient

execution speed. Sort-Matching is implemented by matching two sorted vectors, whose indexes are illustrated in a one-line notation:

$$\begin{aligned} \mathbf{w} : \tau &= (\tau_1 \quad \tau_2 \quad \tau_3 \quad \dots \quad \tau_n) \\ \mathbf{r} : \kappa &= (\kappa_1 \quad \kappa_2 \quad \kappa_3 \quad \dots \quad \kappa_n) \end{aligned} \tag{2.2}$$

The Sort-Matching output is:  $out_{\tau_i} = r_{\kappa_i}$ .

To create a wide range of feature augmentations that combine various styles, we rely on Exact Feature Distribution Mixing (EFDMix) as outlined in Equation 2.3, incorporating interpolated sorted vectors:

$$EFDMix(w, r)_{\tau_i} = w_{\tau_i} + (1 - d)r_{\kappa_i} - (1 - d)\langle w_{\tau_i} \rangle \tag{2.3}$$

We use an instance-specific mixing weight denoted as  $d$ , obtained by sampling from a Beta distribution  $(c, c)$ , where  $c \in (0, \infty)$  serves as a hyperparameter, and  $\langle \cdot \rangle$  represents the stop-gradient operation [107].

**Learnable mixup with style transfer.** For adversarial domain augmentation, we employ feature perturbations, assuming that the perturbations, denoted as  $e$ , follow a multivariate Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ . The parameters of this Gaussian distribution  $(\mu, \sigma)$  are learned using variational inference. The updated latent feature  $h^+$  is obtained by adding the perturbations to the original feature  $h$  with interpolated style through EFDMix (where  $r$  is obtained by shuffling  $h$  along the batch dimension). This is denoted as  $h^+ \leftarrow EFDMix(h, r) + e$ , where  $e$  is sampled from the Gaussian distribution  $\mathcal{N}(\mu, \sigma^2)$ . This approach allows us to create a series of feature augmentations during different training iterations.

Our method also involves blending  $S$  and  $S^+$  via Mixup [108] to achieve intermediate domain interpolations. Specifically, we utilize the uncertainty captured in the perturbations  $(\mu, \sigma^2)$  to predict adjustable parameters  $(a, b)$ , which guide the direction and intensity of domain interpolations.

$$\begin{aligned} h^{+'} &= \lambda \cdot EFDMix(h, r) + (1 - \lambda)h^+ \\ y^+ &= \lambda y + (1 - \lambda)\tilde{y} \end{aligned} \tag{2.4}$$

where  $\lambda \in \text{Beta}(a, b)$  and  $\tilde{y}$  is a smoothed version of  $y$  by a chance of lottery  $\tau$ . The Beta distribution and lottery are computed by a fully connected layer following the Generator. These parameters integrate the uncertainty of generated domains.

**Adversarial domain augmentation.** In the latent space, we propose an iterative training procedure alternating between two phases: a maximization phase, where new data points are learned by computing the inner maximization problem, and a minimization phase, where model parameters are updated according to stochastic gradients of the loss evaluated on the adversarial examples generated from the maximization phase. The fundamental concept here is to iteratively acquire "hard" data points from fictitious target distributions while retaining the essential semantic attributes of the initial data points via adversarial data augmentation [109]:

$$\max_G L(M; S^+) - \beta \|z - z^+\|_2^2 \tag{2.5}$$

where  $L$  represents the cross-entropy loss and involves the creation of  $S^+$  through the perturbations of  $h^+$ ; the second term is the safety constraint that limits the maximum divergence between  $S$  and  $S^+$  in the embedding space.  $z$  ( $z^+$  when  $G$  is activated) denotes the output from the Projection head ( $P$ ), and  $\beta$  is a hyperparameter controlling the maximum divergence. The projection head part of our model transforms the convolutional features into a lower-dimensional feature space  $Z$  suitable for contrastive learning. This is distinct from  $h$ , which denotes the outputs from the convolutional layers.

### 2.2.3 Learning Objective

Our focus is on acquiring cross-domain invariant representations and producing effective domain augmentations  $S^+$ . To achieve this, we utilize the SimCLR [?] contrastive loss and cross-entropy loss as training objectives. First, the cross-entropy loss for the domain alignment network ( $C$ ) is computed as follows:

$$L_{ce} = \mathbb{E} \left[ - \sum_{i=1}^n y_i \log \hat{y}_i \right] \quad (2.6)$$

where  $y_i$  is the ground truth label, and  $\hat{y}_i$  is the prediction by the model.

For the contrastive loss, let  $f$  be the encoder network and  $P$  the projection head. The contrastive loss  $L_{cont}$  for an image  $x_i$  and its corresponding transformed image  $\hat{x}_i$  is defined as follows:

$$L_{cont} = -\mathbb{E} \left[ \log \frac{\exp(\text{sim}(P(f(x_i)), P(f(\hat{x}_i)))/\tau)}{\sum_{j=1}^{2n} \mathbb{1}_{[j \neq i]} \exp(\text{sim}(P(f(x_i)), P(f(x_j)))/\tau)} \right] \quad (2.7)$$

where  $\text{sim}(\cdot, \cdot)$  is the cosine similarity,  $\tau$  is a temperature parameter, and  $2n$  is the total number of augmented images in a batch.

### 2.2.4 Overall Objective

The overall objective function  $L_{total}$  combines both the cross-entropy loss and the contrastive loss, controlled by a balancing parameter  $\alpha$ :

$$L_{total} = L_{ce} + \alpha L_{cont} \quad (2.8)$$

**Table 2.1:** The accuracy of single-source domain generalisation (%) on CIFAR-10-C. Models are trained on CIFAR-10 and evaluated on the CIFAR-10-C.

| Method         | Weather      | Blur         | Noise        | Digital      | Avg          |
|----------------|--------------|--------------|--------------|--------------|--------------|
| ERM [109]      | 67.28        | 56.73        | 30.02        | 62.30        | 54.08        |
| M-ADA [101]    | 75.54        | 63.76        | 54.21        | 65.10        | 64.65        |
| U-SDG [103]    | 76.23        | 65.87        | 53.05        | 68.43        | 65.89        |
| RandConv [102] | 76.87        | 55.36        | 75.19        | 77.51        | 71.23        |
| L2D [98]       | 75.98        | 69.16        | 73.29        | 72.02        | 72.61        |
| MetaCNN [100]  | 77.44        | 76.80        | 78.23        | 81.26        | 78.45        |
| <b>Ours</b>    | <b>89.13</b> | <b>82.94</b> | <b>85.62</b> | <b>84.43</b> | <b>85.53</b> |

By optimizing  $L_{total}$ , CUDGNet is trained to generalize effectively across unseen domains while preserving the semantic information of the original domain and aligning cross-domain representations.

## 2.3 Experimental Results

**Comparison with the state-of-the-art** Tables 2.1, 2.2 exhibit the evaluations of single domain generalisation on CIFAR-10-C and PACS, respectively. The results demonstrate that CUDGNet achieves the highest average accuracy compared to other methods. To be specific, as shown in Table 2.1, there are notable improvements of 11.69%, 6.14%, 7.39%, 3.17% and 7.08% in weather, blur, noise, digital and average categories of CIFAR corruptions respectively. Table 2.2 demonstrates that CUDGNet outperforms all previous methods in the domain of Art Painting except for [110] which has rather imbalanced results and achieves superior average performance. Sketch and Cartoon in contrast to Art painting have huge domain discrepancies compared to Photo (source domain), but still, our model achieves comparable results in these categories compared to the state-of-the-art.

**Comparison with the state-of-the-art** Tables 2.1, 2.2 exhibit the evaluations of single domain generalisation on CIFAR-10-C and PACS, respectively. The results demonstrate that CUDGNet achieves the highest av-

**Table 2.2:** The accuracy of single-source domain generalisation (%) on PACS. Models are trained on photo and evaluated on the rest of the target domains.

| Method        | A            | C            | S            | Avg          |
|---------------|--------------|--------------|--------------|--------------|
| ERM [109]     | 54.43        | 42.74        | 42.02        | 46.39        |
| JiGen [99]    | 54.98        | 42.62        | 40.62        | 46.07        |
| M-ADA [101]   | 58.96        | 44.09        | 49.96        | 51.00        |
| L2D [98]      | 56.26        | 51.04        | 58.42        | 55.24        |
| ALT [110]     | <b>68.50</b> | 43.50        | 53.30        | 55.10        |
| MetaCNN [100] | 54.04        | <b>53.58</b> | <b>63.88</b> | 57.17        |
| <b>Ours</b>   | 59.30        | 50.66        | 62.00        | <b>57.32</b> |

erage accuracy compared to other methods. To be specific, as shown in Table 2.1, there are notable improvements of 11.69%, 6.14%, 7.39%, 3.17% and 7.08% in weather, blur, noise, digital and average categories of CIFAR corruptions respectively. Table 2.2 demonstrates that CUDGNet outperforms all previous methods in the domain of Art Painting except for [110] which has rather imbalanced results and achieves superior average performance. Sketch and Cartoon in contrast to Art painting have huge domain discrepancies compared to Photo (source domain), but still, our model achieves comparable results in these categories compared to the state-of-the-art.

## 2.4 Uncertainty Estimation

**Uncertainty Estimation.** In this section, we evaluate the effectiveness and efficiency of our domain uncertainty score, introduced in Section 2.3, by comparing it with a more computationally intensive Bayesian approach [111]. Our method calculates uncertainty through a single-pass forward operation, which requires approximately 0.15 milliseconds per batch. In contrast, the Bayesian approach relies on repeated sampling, performing 30 passes to compute output variance, which significantly increases the computation time to approximately 5.1 milliseconds per batch.

### 2.4.1 Performance Analysis

To illustrate the performance of our uncertainty estimation method, we present the results on the CIFAR-10-C dataset in Figure 2.2. The outcomes demonstrate that our domain uncertainty score consistently aligns with the Bayesian uncertainty estimation, despite the latter’s higher computational cost. This consistency confirms that our method can achieve similar accuracy in uncertainty estimation without the need for extensive computational resources.

### 2.4.2 Practical Benefits

The substantial reduction in computation time offered by our approach has significant practical benefits, particularly in scenarios requiring real-time or large-scale data processing. Our single-pass method allows for rapid uncertainty estimation, making it suitable for applications where quick decision-making is critical, such as autonomous driving, medical diagnosis, and real-time surveillance systems.

### 2.4.3 Robustness and Reliability

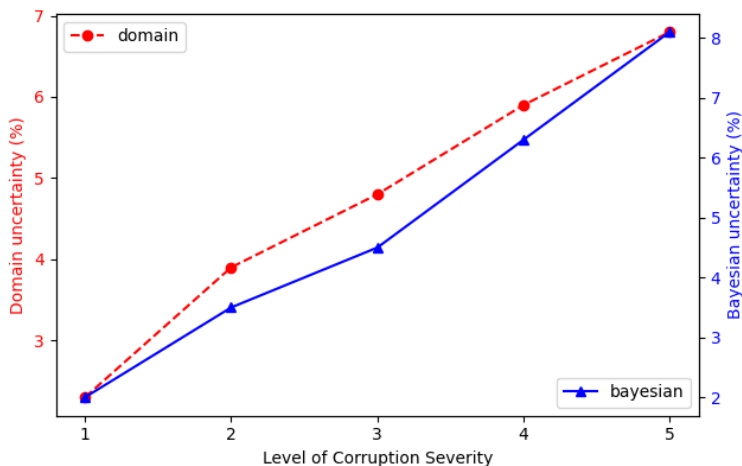
Moreover, the strong alignment between our method and the Bayesian approach underscores the robustness and reliability of our domain uncertainty score. This reliability is essential for various tasks, including out-of-distribution detection and model confidence evaluation. The ability to provide accurate uncertainty estimates ensures that the model can effectively handle unexpected inputs and maintain performance across different domains.

In conclusion, our domain uncertainty score presents a highly efficient and reliable alternative to traditional Bayesian uncertainty estimation methods. By achieving comparable accuracy with a fraction of the computational

**Table 2.3:** Ablation study of the key components of CUDGNet

| Method                 | Weather | Blur  | Noise | Digital | Avg   |
|------------------------|---------|-------|-------|---------|-------|
| Baseline               | 63.43   | 55.61 | 31.92 | 61.01   | 53.01 |
| + G                    | 72.19   | 61.37 | 51.58 | 62.71   | 61.96 |
| + TC                   | 79.22   | 75.44 | 71.46 | 75.14   | 75.31 |
| + Style transfer       | 87.81   | 82.10 | 82.05 | 80.76   | 83.18 |
| + Contrastive learning | 89.13   | 82.94 | 85.62 | 84.43   | 85.53 |

cost, our method stands out as a practical solution for real-world applications. The alignment of our results with those of the Bayesian approach further validates the effectiveness of our uncertainty estimation technique, highlighting its value in the field of domain generalization and image recognition.



**Figure 2.2:** Estimation of Uncertainty CIFAR-10-C. Our domain uncertainty prediction aligns with Bayesian uncertainty, while our approach is significantly faster.

**Ablation Study** As depicted in Table 2.3, the incorporation of various components into our model significantly enhances its performance compared to the baseline with the adversarially augmented generator. This ablation study highlights the effectiveness of each individual component in improving



the model’s generalization capabilities.

**Transformation Component.** Incorporating the Transformation Component leads to significant improvements over the baseline model. This improvement underscores the importance of slightly augmenting the diversity of the source domain and altering the image structure through fractals before the adversarial min-max optimization of the generator. By introducing these transformations, the model is exposed to a broader variety of features and patterns, enabling it to generalize better when confronted with previously unseen domains. This step is crucial as it helps the model learn more robust and invariant features, contributing to its overall performance.

**Style Transfer.** Further enhancement is observed with the addition of the style transfer component, resulting in a performance boost of 7.87%. This substantial improvement indicates that style transfer is a powerful tool for domain generalization. By altering the style of images, the model becomes more adept at recognizing the underlying content regardless of stylistic variations. This capability is particularly valuable in scenarios where the target domain may exhibit different visual styles from the source domain. The style transfer component effectively diversifies the training data, making the model more resilient to domain shifts and stylistic discrepancies.

**Contrastive Loss.** The integration of contrastive loss marks another significant milestone in our model’s performance. With this addition, we achieve a new state-of-the-art performance, boasting an average performance score of 85.53%. Contrastive loss helps in learning domain-invariant representations by encouraging the model to minimize the distance between similar samples while maximizing the distance between dissimilar ones. This approach ensures that the learned features are more discriminative and less sensitive to domain-specific variations. The success of this component demonstrates its crucial role in enhancing the model’s ability to generalize across different domains.

In summary, the ablation study presented in Table 2.3 clearly demonstrates

the effectiveness of each component in improving the model’s robustness and generalization capabilities. The Transformation Component, style transfer, and contrastive loss each contribute uniquely to the overall performance, culminating in a model that sets a new benchmark in single domain generalization for image recognition. These findings validate our approach and highlight the potential of our model to handle a wide range of domain variations, ultimately advancing the field of out-of-distribution robustness in image recognition.

## **2.5 Conclusion**

This chapter explored the challenge of out-of-distribution robustness in image recognition, specifically focusing on single domain generalization. We introduced the Contrastive Uncertainty Domain Generalisation Network (CUDGNet), a novel model designed to enhance performance on unseen domains by augmenting the source capacity in both input and label spaces through a fictitious domain generator. The model also leverages contrastive learning to achieve domain invariant representations for each class.

Our approach demonstrates significant domain expansion capabilities while avoiding representation collapse. Through extensive experiments on two Single Source Domain Generalisation (SSDG) datasets, we showed that CUDGNet surpasses the current state-of-the-art single-DG methods by up to 7.08%. Additionally, CUDGNet offers efficient uncertainty estimation at inference time via a single forward pass through the generator subnetwork, highlighting its practical applicability and effectiveness in real-world scenarios. This work contributes to advancing the robustness of image recognition models when faced with diverse and unfamiliar domains, underscoring the potential of CUDGNet in achieving reliable and generalizable performance.

# Chapter 3

## Robustness in Medical Imaging

### 3.1 Introduction

Medical image analysis (MedIA) has become an integral part of modern medical practice, playing a crucial role in disease diagnosis, prognosis, and treatment planning. Recent advancements in deep learning (DL) have significantly accelerated progress in this field. However, applying DL models to real-world medical imaging scenarios presents substantial challenges, primarily due to the models' limited ability to generalize effectively across the distributional gap between training and testing samples—a phenomenon known as domain shift. Addressing this issue, researchers have dedicated substantial efforts to developing various DL approaches to ensure robust performance when confronted with unknown and out-of-distribution data distributions.

This PhD thesis aims to make a substantial contribution to enhancing out-of-distribution robustness in medical imaging, particularly focusing on applications for COVID-19 detection. Through comprehensive research and analysis, this work explores the development and evaluation of innovative methodologies designed to improve the reliability and effectiveness of medical imaging techniques in identifying COVID-19 cases, even in scenarios beyond the scope of the training data. By tackling the challenges associated with out-of-distribution scenarios, this thesis aspires to advance the state-of-the-art in medical imaging, thereby facilitating more accurate and depend-

able diagnoses amid the ongoing COVID-19 pandemic.

Out-of-distribution robustness has become a pivotal area within deep learning, particularly when the capacity to generalize across diverse domains is paramount. As highlighted above, it is especially critical in medical image analysis (MedIA), which is characterized by highly heterogeneous data. To comprehensively understand the unique challenges associated with domain generalization in MedIA, it is crucial to consider the following factors impacting Out-of-Distribution Robustness for Medical Image Analysis:

- **Variability in Medical Imaging:** Variability in medical imaging stems from differences and inconsistencies during the data acquisition process [112]. These variations may be external, resulting from the use of different modalities, protocols, scanner types, and patient demographics across various healthcare facilities. Internal variability can also occur within a controlled setting (e.g., the same scanner or healthcare facility) due to factors such as hardware aging, software parameter variations, and human error (e.g., motion artifacts).
- **Complex and High-Dimensional Data:** Medical images often exhibit significant complexity and high dimensionality, with multiple channels or sequences. Many datasets range from thousands of pixels to gigapixels [113] and span from 2D to 5D dimensions [114]. This complexity poses challenges in identifying and extracting domain-invariant features that can generalize effectively across diverse domains.
- **Challenging Data Acquisition, Organization, and Labeling:** Acquiring, organizing, and labeling data in medical imaging is arduous. Large-scale, diverse, and labeled datasets are difficult to obtain due to the high costs of data acquisition, privacy concerns, data sharing restrictions, and the labor-intensive nature of manual annotation by medical professionals. Additionally, ensuring quality assurance is challenging, as medical images are prone to noise and artifacts, such as patient motion, scanner imperfections, and hardware or software

limitations.

- **Model Interpretability, Safety, and Privacy:** In MedIA, ensuring model interpretability, safety, and compliance with regulatory and ethical standards is paramount. Robustness against adversarial examples and out-of-distribution samples is crucial to prevent adverse effects on patient care. Furthermore, enabling privacy-preserving data sharing and collaboration in multi-center contexts adds complexity to the implementation of domain generalization techniques.

This PhD thesis aims to address these challenges by developing robust methodologies for out-of-distribution generalization in medical imaging, with a particular focus on enhancing the reliability and effectiveness of COVID-19 detection. Through this work, we seek to contribute to the advancement of MedIA, ensuring more accurate and dependable medical diagnoses in the face of diverse and unforeseen data distributions.

## 3.2 The COV19-CT-DB Database

### 3.2.1 Introduction to the Database

To develop a robust and accurate AI model for medical imaging, particularly for detecting COVID-19 from CT scan images, it is paramount to have a large and representative training dataset. Such a dataset ensures that the model can generalize well to various real-world scenarios and patient demographics. In response to this critical need, we introduce the COV19-CT-DB (COVID-19 Computed Tomography Database), a comprehensive and extensive annotated database of chest CT scans aimed at enhancing COVID-19 classification.

The COV19-CT-DB database is a significant contribution to the medical imaging community, addressing the scarcity of large-scale annotated datasets necessary for training deep learning models. This database com-

prises 7,750 3-D CT scans collected from multiple hospitals, ensuring a diverse representation of cases. Of these, 1,650 scans are annotated as COVID-19 cases, while 6,100 scans are classified as non-COVID-19 cases. The inclusion of such a large number of non-COVID-19 cases is crucial for developing models that can accurately distinguish between COVID-19 and other conditions, thereby reducing false positives.

Each 3-D CT scan in the database contains a varying number of slices, ranging from 50 to 700, resulting in a total of approximately 2,500,000 CT slices. This extensive variability in the number of slices per scan reflects real-world clinical settings, where the number of slices can vary significantly based on the scanning protocol and patient condition. By incorporating this variability, the database provides a rich and challenging dataset for training robust AI models.

Furthermore, part of the COV19-CT-DB database has already been successfully utilized in a recently held competition, demonstrating its practical utility and relevance to the research community [115]. The entire database is now being made available to researchers for further investigation and development of AI models. This open access to the database aims to foster collaboration and accelerate advancements in the field of medical image analysis, particularly in the context of COVID-19.

The availability of such a comprehensive dataset is expected to drive significant improvements in the performance and generalization of AI models for COVID-19 detection. Researchers can leverage the COV19-CT-DB to train and validate their models, ensuring that these models are not only accurate but also robust to various real-world conditions and patient populations. By contributing to the broader research community, this database supports the ongoing efforts to combat the COVID-19 pandemic through advanced AI-driven diagnostic tools.

### 3.2.2 Data Collection and Processing

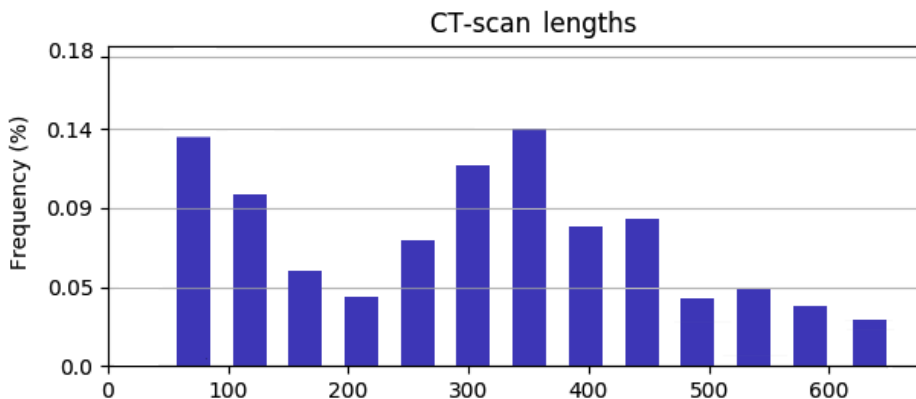
Data collection for COV19-CT-DB was conducted from September 1, 2020, to November 30, 2021. The dataset comprises 1,650 COVID-19 and 6,100 non-COVID-19 chest CT scan series, representing a large number of patients (more than 1,150) and subjects (more than 2,600). Due to the anonymization procedure, no specific patient and subject numbers can be reported.

The data was collected using the Siemens Somatom Emotion 16-section CT scanner at AHEPA Hospital's Emergency Department. Scans covered from the upper thoracic outlet to the diaphragm of the patients. Key parameters included detector collimation widths of 6 x 0.5 mm, tube voltage of 130 kV, and 114 mA. Images were reconstructed with a slice thickness of 0.75 mm in a lung window, with patients in a supine position.

HRCT (High Definition Computed Tomography) was used due to its ability to produce detailed lung images, essential for diagnosing and monitoring COVID-19 pneumonia. After HRCT, the data was stored in the Image Archiving and Processing System of the Clinical Radiology Department at AHEPA Hospital and subsequently anonymized using the DicomCleaner program. This process involved overwriting each patient's name, ID, and other non-secure private attributes, ensuring that the HRCTs were cleared and anonymized.

### 3.2.3 Annotation Process

Each CT slice was annotated by four experienced medical experts: two radiologists and two pulmonologists, each with over 20 years of experience. The experts identified and labeled COVID-19-related abnormalities, such as ground-glass opacities and consolidations, using a standardized protocol. The annotations demonstrated a high degree of agreement (around 98%), ensuring consistency and accuracy.



**Figure 3.1:** Histogram of CT scan lengths in COV19-CT-DB

### 3.2.4 Database Composition

Figure 3.2 displays slices from a CT scan series of a non-COVID-19 case. Similarly, Figure 3.3 presents slices from a CT scan series of a COVID-19 case. Figure 3.4 illustrates four CT scan slices: two from a non-COVID-19 scan on the left and two from a COVID-19 scan on the right. Bilateral ground-glass opacities are particularly evident in the lower lung lobes of the COVID-19 slices.

The database is divided into training, validation, and test sets. The training set includes 1,991 CT scans in total, with 882 labeled as COVID-19 and 1,109 labeled as non-COVID-19. The validation set comprises 484 CT scans, with 215 labeled as COVID-19 and 269 as non-COVID-19. The test set consists of 5,281 CT scans, with 564 labeled as COVID-19 and 4,717 labeled as non-COVID-19.

Some CT series from the same individual were taken at different times. To ensure there was no data leakage from the training to the test set, we compared each 3-D CT scan in the test set with each 3-D CT scan in the training set. Initially, we compared each 3-D CT scan in the test set with all 3-D CT scans in the training set that had the same length in terms of



the number of CT slices. Subsequently, we compared each 3-D CT scan in the test set with all 3-D CT scans in the training set that did not have the same length, making the comparison over the minimum CT scan length, i.e., the first 50 slices of each 3-D CT scan. We found no 3-D CT scan in the test set that was identical or nearly identical to any 3-D CT scan in the training set.

Finally, Table 3.1 summarizes the main attributes of COV19-CT-DB as described above.

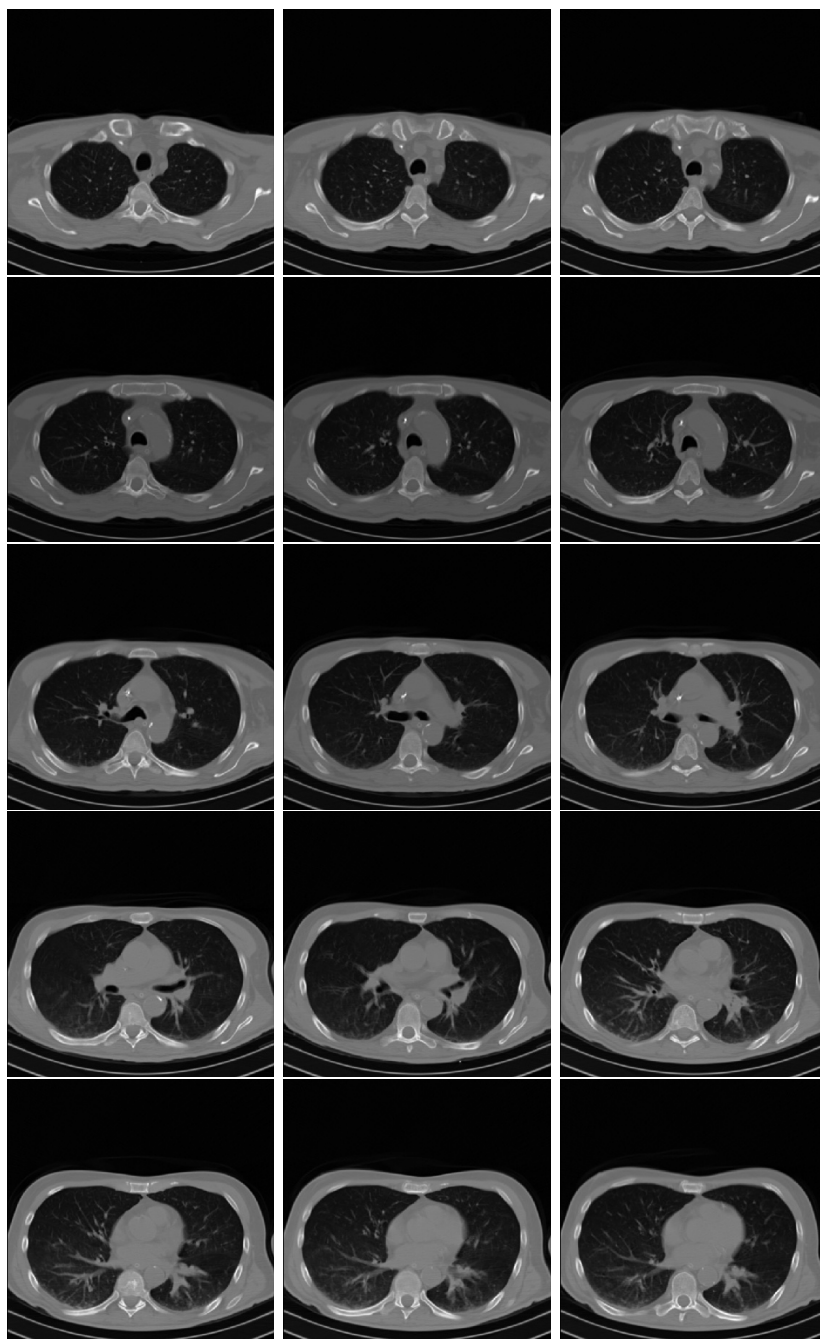
### 3.2.5 Ethical Considerations

The collection and use of the data were conducted ethically, with appropriate consideration for patient privacy and informed consent. Anonymization ensured that patient data was protected, preventing misuse or unauthorized disclosure.

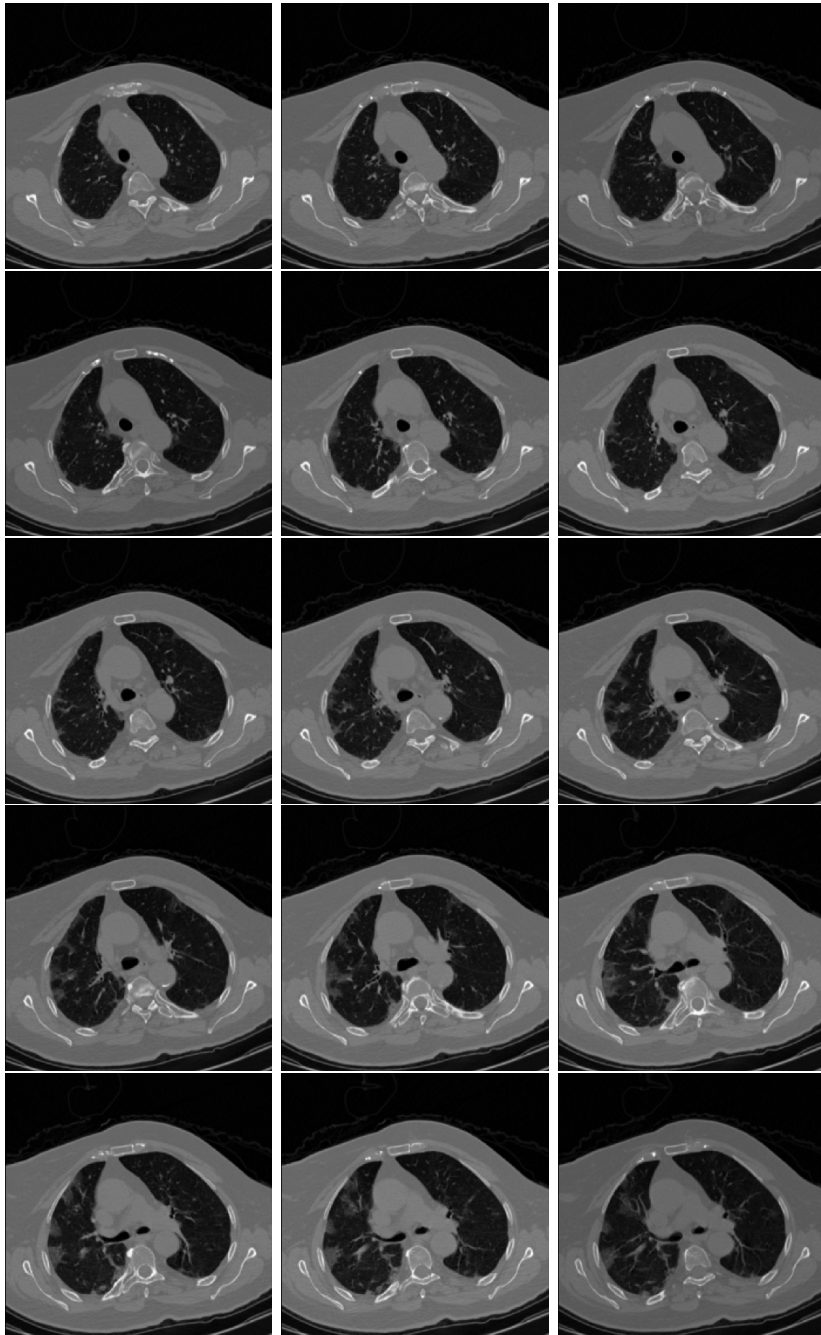
Finally, Table 3.1 summarizes the main attributes of COV19-CT-DB that are presented above.

**Table 3.1:** Attributes of the COV19-CT-DB

| Attribute               | Description  |
|-------------------------|--|
| total # of CT scans     | 1,661 COVID<br>6,095 non-COVID   |
| total # of slices       | 724,273 from CT scans of COVID<br>1,775,727 from CT scans of non-COVID |
| # of slices per CT scan | 50 - 700   |
| # Patients              | >1150  |
| # Subjects              | >2600  |
| slice image resolution  | 512 × 512  |
| # Annotators            | 4 medical experts<br>(2 radiologists & 2 pulmonologists)               |



**Figure 3.2:** Slices from a non COVID-19 CT scan in COV19-CT-DB



**Figure 3.3:** Slices from a COVID-19 CT scan in COV19-CT-DB



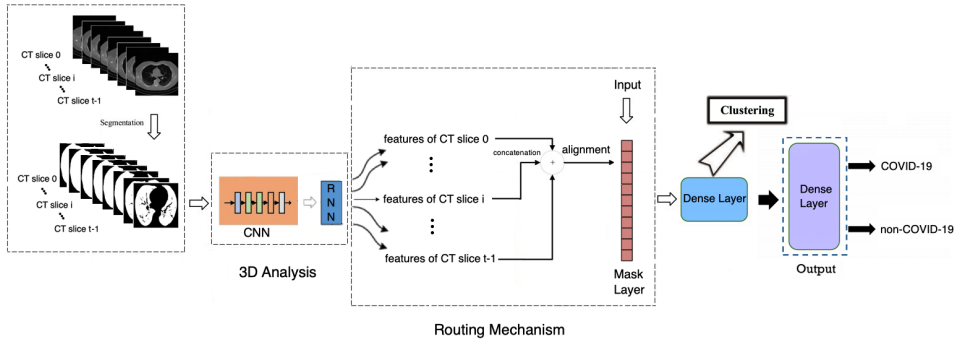
**Figure 3.4:** Four CT scan slices in COV19-CT-DB, the top 2 from a non-COVID-19 scan and the bottom 2 from a COVID-19, including bilateral ground glass regions in lower lung lobes.

### 3.3 RACNet: The Proposed Architecture

This section elaborates on the characteristics of the input data used for diagnosing COVID-19 from chest CT scans. The input data consists of a series of chest CT slices, which form a 3-D signal. Each slice is a 2-D image, and the number of slices varies across different CT scans. While a 3-D CNN architecture, such as a 3-D ResNet, can process these 3-D signals effectively, handling the variation in the number of slices per CT scan presents a significant challenge. To address this, various ad-hoc methods can be employed, such as selecting a fixed input length and either discarding extra slices or duplicating slices when there are fewer than the required number.

In this PhD thesis, we introduce a novel architecture named RoutingAlign-ClusterNet (RACNet), which integrates both CNN and RNN components, diverging from the conventional purely 3-D CNN approach. RACNet consists of three main components: the 3D Analysis component, the Routing component, and the Classification component. Initially, to standardize the input data, all CT scans are padded to have a uniform length  $t$ , meaning each scan is adjusted to have  $t$  slices. For instance, a CT scan with only 50 slices would be expanded to 700 slices through duplication.

Our model processes input data in two distinct modes. In the first mode, each 2-D slice is segmented to isolate the lung regions, and these segmented images are then fed into the CNN. In the second mode, the entire, unsegmented 2-D slices are used as input to the CNN. Both methods are thoroughly evaluated in the experimental studies included in this thesis. The subsequent sections provide a detailed description of each component of the RACNet architecture.



**Figure 3.5:** The proposed Pipeline: A 3-D input composed of up to  $t$  chest CT slices is processed for COVID-19 diagnosis. A CNN-RNN architecture performs 3-D analysis, while a routing mechanism with an ‘alignment’ step and a Mask Layer handles the varying input length  $t$ . A dense layer precedes the output layer that provides the COVID-19 diagnosis; the neuron outputs of the dense layer are further analyzed through clustering to derive a latent variable model and a related set of anchors that provide additional insights into the decision-making process.

### 3.3.1 3-D Analysis Component

The input data are processed by the 3D Analysis component of RACNet, which consists of a Convolutional Neural Network (CNN) followed by a Recurrent Neural Network (RNN). The CNN is responsible for performing local analysis on each 2-D slice, focusing primarily on extracting features from the lung regions. This feature extraction aims to replicate the diagnostic process used by medical experts who annotate the data based on the entire 3-D CT scan series. After the CNN has extracted the features from each slice, the RNN sequentially processes these features from the entire 3-D CT scan, starting from slice 0 and continuing to slice  $t - 1$ . This sequential analysis by the RNN allows for the temporal and spatial relationships between slices to be taken into account, enhancing the overall diagnostic accuracy. This process is depicted in Figure 3.5, where  $t$  represents the maximum number of slices available in the chest CT scans.

### 3.3.2 Routing Component

As depicted in Figure 3.5, the RNN features corresponding to each CT slice (from 0 to  $t - 1$ ) are inputs to the Routing Component of RACNet. These features are concatenated and fed into the Mask Layer. The original input series length  $l$  is passed to the Mask Layer to guide the routing process.

The Mask Layer dynamically selects RNN outputs based on the input length  $l$ , retaining the values of the selected outputs and zeroing out the rest. This dynamic routing procedure is illustrated in Figure 3.5. During the training of RACNet, the routing mechanism selects the RNN outputs as indicated by the input series length  $l$ .

Two methods can be employed for this selection process:

- **Method a: Selecting the first  $l$  RNN outputs** - This approach involves simply selecting the first  $l$  features from the sequence of RNN outputs, corresponding to the original CT slices.
- **Method b: Performing an 'alignment' step** - This method involves placing the original  $l$  RNN outputs in equidistant positions within the range  $[0, t - 1]$  and placing the remaining outputs in the intervening positions.

The Mask Layer then routes the selected RNN outputs to the Classification module.

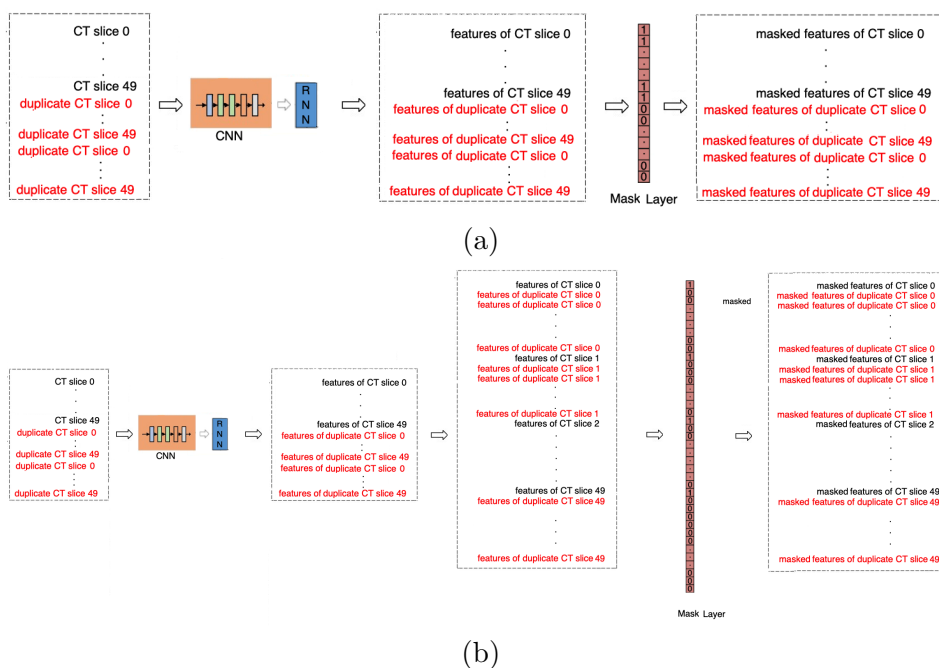
**The 'alignment' step** For instance, consider a scenario with a maximum input length of 700 CT scan slices. If a CT scan comprises 50 slices, it is padded with 650 duplicate slices to achieve a total of 700 slices. During training, all 700 slices are processed by the CNN-RNN.

Without alignment, the Mask Layer zeroes out the features of the 650 duplicate slices and retains the first 50 features. This is illustrated in Figure

3.6 (a).

With alignment, features from the original 50 slices are placed in equidistant positions within the range  $[0, 699]$ , and the duplicate slice features are placed in the intervening positions. The Mask Layer then zeroes out the features of the duplicate slices, retaining the features of the original slices and routing them to the Classification module. This is illustrated in Figure 3.6 (b).

The alignment step enhances training efficiency by ensuring that the weights are updated with similar slice information, making the training process more effective, particularly when dealing with limited data.



**Figure 3.6:** The Routing Mechanism: (a) without and (b) with the 'alignment' step



### 3.3.3 Classification Component

The concatenated RNN outputs are fed into the Classification Component of RACNet, which consists of a dense layer followed by an output layer utilizing a softmax activation function for the final COVID-19 diagnosis. The dense layer is designed to extract high-level information from the concatenated RNN outputs. During the training process, only the weights that connect the neurons of the dense layer with the routed RNN outputs are updated. This dynamic routing principle ensures that non-routed RNN outputs are ignored, keeping their corresponding weights constant. This targeted update mechanism allows the model to focus on the most relevant features for the diagnosis task, thereby improving the accuracy and robustness of the final prediction.

### 3.3.4 Latent Variable Analysis and Anchor Set Generation

Following the training of RACNet, we proceed to extract and analyze the neuron outputs of the dense layer through a clustering approach. These latent variables encapsulate high-level semantic information that is crucial for the final classification. By discarding the output layer and performing an unsupervised analysis, we aim to generate a representation that offers deeper insights into the decision-making process of the model.

To achieve this, we input a training dataset into the trained RACNet architecture, extracting the dense layer neuron outputs for each 3-D CT scan input  $k$ . These outputs form vectors denoted as  $\mathbf{v}(k)$ . We then employ a clustering algorithm, such as k-means++, to generate a concise representation of these vectors by minimizing the following criterion:

$$\hat{Q}_{k\text{-means}} = \arg \min_Q \sum_{i=1}^M \sum_{\mathbf{v} \in V} \mathbf{v} - \mu_i^2 \quad (3.1)$$

where  $\mu_i$  represents the mean of the  $\mathbf{v}$  values within cluster  $i$ . Each cluster

center  $\mathbf{c}(i)$  is subsequently computed, forming a concise representation set  $\mathcal{C}$ :

$$\mathcal{C} = \{\mathbf{c}(i) \in \mathbb{R}^L, i = 1, \dots, M\} \quad (3.2)$$

Medical experts can then review the CT scan inputs corresponding to these cluster centers, providing additional semantic information to the representation. The generated set  $\mathcal{C}$  serves as an anchor model for diagnosing new cases. When testing RACNet on a new CT scan, the corresponding  $\mathbf{v}$  vector is extracted, and its Euclidean distance to each cluster center in  $\mathcal{C}$  is computed. The new case is associated with the nearest cluster center and labeled accordingly.

This method of latent variable extraction and anchor set generation significantly aids in the efficient diagnosis of COVID-19 by enabling the comparison of  $L$ -dimensional vectors and selecting the minimum distance. Moreover, this approach provides confidence levels for the diagnosis and facilitates efficient training updates with new datasets. Medical centers can share their best-performing networks and anchor sets, continuously improving the data-driven representations and diagnostic capabilities across different institutions.

### **3.4 Experimental Study**

This section describes a comprehensive set of experiments conducted to evaluate the performance of the proposed RACNet approach for COVID-19 detection across various databases. The experiments are designed to validate the efficacy of RACNet compared to existing state-of-the-art methods and to analyze the contribution of each component within the RACNet architecture.

### 3.4.1 Training RACNet on the COV19-CT-DB Database

Initially, RACNet was trained on the COV19-CT-DB database, as described in Section 3.2. We compared RACNet’s performance against several other neural network architectures, including standard 2D and 3D CNNs. Additionally, we conducted an ablation study to verify the contribution of each component of RACNet (3D Analysis, Routing, and Classification). The results demonstrated that RACNet outperformed other architectures, highlighting the effectiveness of its design.

### 3.4.2 Application and Retraining on Additional Databases

To further validate RACNet, we applied and retrained it on five additional databases, comparing its performance to state-of-the-art methods specifically tailored for COVID-19 detection. The databases used in these experiments include:

**COV19D-ICCV2021** This database was shared during the COV19D Competition at the “AI-enabled Medical Image Analysis Workshop and Covid-19 Diagnosis Competition” held in conjunction with the International Conference on Computer Vision (ICCV) 2021 [115–118]. It includes 1,405 COVID-19 and 4,066 non-COVID-19 3D CT scans, with 707 COVID-19 and 845 non-COVID-19 scans in the training set, and 165 COVID-19 and 209 non-COVID-19 scans in the validation set.

**COV19D-ECCV2022** This database was utilized in the COVID-19 Detection Challenge at the 2nd COV19D Competition held in conjunction with the European Conference on Computer Vision (ECCV) 2022 [115–120]. It comprises 1,550 COVID-19 and 5,044 non-COVID-19 3D CT scans, with 882 COVID-19 and 1,110 non-COVID-19 scans in the training set, and 215 COVID-19 and 289 non-COVID-19 scans in the validation set.

**CC-CCII** We also utilized the Clean CC-CCII database [121, 122], which includes 3D CT scans of three classes: novel coronavirus pneumonia (NCP), common pneumonia (CP), and Normal. The training partition consists of 3,195 3D CT scans (1,213 NCP, 1,210 CP, and 772 Normal), and the test partition consists of 798 3D CT scans (302 NCP, 303 CP, and 193 Normal).

**MosMedData** The MosMedData database [123], annotated for COVID-19/non-COVID-19 diagnosis, contains 1,110 3D CT scans. The COVID class includes 856 scans, and the Normal class includes 254 scans. The training set consists of 601 COVID-19 and 178 non-COVID-19 scans, while the testing set comprises 256 COVID-19 and 76 non-COVID-19 scans.

**CT-image DB** This database [124], annotated for COVID-19/non-COVID-19 diagnosis, contains 2D CT scan slices (408 non-COVID-19 and 349 COVID-19). We augmented these slices using random rotation, noise addition, and horizontal flips to create 3D CT scans. The training set consists of 279 COVID-19 and 326 non-COVID-19 slices, while the test set includes 70 COVID-19 and 82 non-COVID-19 slices.

### 3.5 RACNet Training: Implementation Details

In this section, we provide a comprehensive overview of the implementation details and training procedures used for RACNet in the context of COVID-19 detection from chest CT scans. This includes the architecture specifications, data preprocessing steps, training protocols, hyperparameter tuning, and computational resources employed.

### 3.5.1 Architecture Specifications

RACNet is composed of three primary components: the 3D Analysis Component, the Routing Component, and the Classification Component.

**3D Analysis Component** This component consists of a 3D Convolutional Neural Network (CNN) followed by a Recurrent Neural Network (RNN). The 3D CNN is responsible for performing local analysis on each 2D slice of the CT scan, extracting relevant features. The RNN sequentially processes these features across the slices, capturing temporal dependencies.

**Routing Component** The Routing Component uses dynamic routing principles to select and retain relevant features from the RNN outputs based on the input length. This includes a Mask Layer that dynamically selects outputs based on the input sequence length  $l$ .

**Classification Component** The final component includes a dense layer followed by a softmax output layer, which performs the final classification for COVID-19 diagnosis. The dense layer aggregates high-level features from the RNN outputs, and the softmax layer provides probabilistic outputs for classification.

### 3.5.2 Data Preprocessing

To ensure uniformity and compatibility across different databases, the following preprocessing steps were applied:

- **Resizing:** All CT scan slices were resized to a fixed resolution of 224x224 pixels.

- **Normalization:** Pixel intensity values were normalized to a range of  $[0, 1]$  to standardize the input data.
- **Augmentation:** Data augmentation techniques, such as random rotations, horizontal flipping, and adding Gaussian noise, were employed to increase the diversity of the training data and reduce overfitting.
- **Padding:** For CT scans with fewer slices than the maximum sequence length  $t$ , slices were duplicated to reach the required length. For scans with more slices, excess slices were removed.

### 3.5.3 Training Protocols

The training process of RACNet followed a systematic protocol to ensure robust model performance:

- **Loss Function:** The categorical cross-entropy loss was used for optimizing the model.
- **Optimizer:** The Adam optimizer was chosen for its adaptive learning rate capabilities, with an initial learning rate of  $1 \times 10^{-4}$ .
- **Batch Size:** A batch size of 16 was used to balance memory usage and training speed.
- **Epochs:** The model was trained for 100 epochs, with early stopping based on the validation loss to prevent overfitting.
- **Learning Rate Schedule:** A learning rate scheduler was implemented to reduce the learning rate by a factor of 0.1 if the validation loss did not improve for 10 consecutive epochs.

### 3.5.4 Hyperparameter Tuning

Extensive hyperparameter tuning was conducted to optimize RACNet’s performance. This included:

- **Grid Search:** A grid search approach was used to explore different combinations of hyperparameters, such as learning rate, batch size, and the number of layers in the 3D CNN and RNN.
- **Cross-Validation:** Five-fold cross-validation was employed to ensure that the selected hyperparameters generalize well across different subsets of the training data.
- **Regularization:** Dropout layers with a dropout rate of 0.5 were incorporated to prevent overfitting, and L2 regularization was applied to the dense layer.

### 3.5.5 Computational Resources

The training of RACNet was performed using high-performance computational resources to handle the extensive data and complex computations:

- **Hardware:** Training was conducted on NVIDIA Tesla V100 GPUs, each with 32 GB of memory.
- **Software:** The implementation was carried out using TensorFlow and Keras libraries for neural network modeling and training.
- **Environment:** The experiments were conducted in a Linux-based environment with CUDA and cuDNN enabled for GPU acceleration.

### 3.5.6 Evaluation Metrics

To assess the performance of RACNet, the following evaluation metrics were used:

- **Accuracy:** The overall accuracy of the model in correctly classifying COVID-19 and non-COVID-19 cases.
- **Precision, Recall, F1-Score:** These metrics were calculated to evaluate the model's performance in terms of precision, recall, and the harmonic mean of precision and recall.
- **ROC-AUC:** The Area Under the Receiver Operating Characteristic Curve was used to measure the model's ability to distinguish between classes.
- **Confusion Matrix:** A confusion matrix was generated to provide a detailed breakdown of true positives, false positives, true negatives, and false negatives.

For the implementation of RACNet, we employed EfficientNetB0 as the CNN model, with a global average pooling layer, batch normalization, and dropout (keep probability of 0.8). The RNN component consisted of a single-directional GRU layer with 128 units, followed by a dense layer with 128 hidden units.

During training, we used a batch size of 5 and an input length 't' of 700 slices, each resized from its original size of  $512 \times 512$  to  $256 \times 256$  pixels. The loss function employed was cross entropy, optimized using the Adam optimizer with a learning rate of  $10^{-4}$ . K-means clustering was performed with values of  $k$  ranging from 2 to 25. Training computations were carried out on a Tesla V100 32GB GPU.



## 3.6 Experimental study

### 3.6.1 Experiments & Ablation Study on COV19-CT-DB

In this section, we present comprehensive experiments conducted on the COV19-CT-DB database to evaluate the performance of RACNet compared to other state-of-the-art networks.

#### Performance Comparison

We compared RACNet’s performance with several established models, including 3-D CNN and CNN-RNN architectures, trained and tested on the COV19-CT-DB dataset. Specifically, we employed pre-trained models such as 3-D ResNet-50 [125] and MedicalNet [126]. Table 3.2 summarizes the performance metrics of these models alongside RACNet.

**Table 3.2:** Performance comparison between RACNet and other state-of-the-art networks on the test set of COV19-CT-DB database (non-segmented data)

| Method             | COVID Accuracy | non-COVID Accuracy | F1 Score    |
|--------------------|----------------|--------------------|-------------|
| 3D ResNet-50 [125] | 0.74           | 0.80               | 0.82        |
| MedicalNet [126]   | 0.78           | 0.83               | 0.86        |
| RACNet             | <b>0.82</b>    | <b>0.86</b>        | <b>0.90</b> |

RACNet consistently outperformed these models in terms of accuracy and F1 score, demonstrating its effectiveness in COVID-19 detection from chest CT scans.

#### Ablation Study

To analyze the contribution of each component in RACNet, we conducted ablation studies varying different aspects of the architecture:

- **Routing Mechanism:** Introducing a mask in the routing mechanism to filter out redundant information significantly enhanced overall performance.
- **Alignment Step:** Including an alignment step improved feature alignment across different CT scan lengths, contributing to better diagnostic accuracy.
- **CNN Architectures:** We evaluated various CNN backbones, including EfficientNetB0, ResNet-50, DenseNet-121, and a model with 3-D convolutions, to assess their impact on performance.
- **Dense Layer Units:** Different configurations of the dense layer were tested, with varying numbers of hidden units.

Table 3.3 presents the results of these ablation studies, showcasing the performance under different configurations.

**Table 3.3:** Ablation Study: Performance comparison on the test set of COV19-CT-DB database (non-segmented data)

| Configuration  | COVID Accuracy | non-COVID Accuracy | F1 Score    |
|--|----------------|--------------------|-------------|
| 64 units in dense layer                              | 0.79           | 0.83               | 0.86        |
| 16 units in dense layer                              | 0.78           | 0.82               | 0.85        |
| 3-D conv instead of CNN-RNN                          | 0.79           | 0.84               | 0.87        |
| ResNet-50 as CNN                                     | 0.80           | 0.85               | 0.88        |
| DenseNet-121 as CNN                                  | 0.79           | 0.84               | 0.87        |
| without 'alignment'                                  | 0.80           | 0.85               | 0.88        |
| without mask   | 0.78           | 0.84               | 0.87        |
| <b>EfficientNetB0, GRU, 128 units in dense layer</b> | <b>0.82</b>    | <b>0.86</b>        | <b>0.90</b> |

The ablation study results demonstrate that the combination of EfficientNetB0 as the CNN backbone, GRU in the routing mechanism, and 128 units in the dense layer achieved the highest COVID accuracy, non-COVID accuracy, and F1 score on the COV19-CT-DB dataset.

These experiments confirm the efficacy of RACNet in COVID-19 detection from chest CT scans and highlight the importance of each architectural component in achieving superior performance.

### **3.6.2 Experiments on COV19D ICCV & ECCV Competitions**

In this section, we analyze the performance of RACNet for COVID-19 detection using segmented CT scans from the COV19D-ICCV2021 and COV19D-ECCV2022 databases. We compare RACNet’s performance with the best-performing methods from the respective competitions.

Until this point, the results presented were based on experiments where no segmentation was applied to the 3-D CT scan inputs. This approach aimed to avoid bias introduced by specific lung segmentation methods on the analysis and results obtained.

From this section onwards, we evaluate RACNet’s performance using segmented CT scans and compare it against the top-performing methods from the ICCV 2021 and ECCV 2022 competitions on COV19D-ICCV2021 and COV19D-ECCV2022 databases, respectively.

#### **Performance Comparison**

Table 3.4 provides a detailed comparison of RACNet’s performance with the state-of-the-art methods from the competitions. The evaluation metrics include macro F1 score, COVID detection accuracy, and non-COVID detection accuracy.

**Table 3.4:** Performance comparison between RACNet and the state-of-the-art on the test set of COV19D-ICCV2021 and COV19D-ECCV2022 databases of the respective ICCV and ECCV Competitions; F1 Score presented in %

| Databases       | Methods              | F1           |              |              |
|-----------------|----------------------|--------------|--------------|--------------|
|                 |                      | Macro        | COVID        | non-COVID    |
| COV19D-ICCV2021 | ACVLab [127]         | 88.74        | 80.63        | 96.84        |
|                 | SenticLab.UAIC [128] | 90.06        | 82.96        | 97.17        |
|                 | FDVTS_COVID [129]    | 90.43        | 83.60        | 97.27        |
|                 | <b>RACNet</b>        | <b>93.83</b> | <b>93.62</b> | <b>94.04</b> |
| COV19D-ECCV2022 | MDAP [130]           | 87.87        | 78.80        | 96.95        |
|                 | FDVTS [131]          | 89.11        | 80.92        | 97.31        |
|                 | ACVLab [132]         | 89.11        | 80.78        | 97.45        |
|                 | <b>RACNet</b>        | <b>95.06</b> | <b>94.18</b> | <b>95.95</b> |

From Table 3.4, it is evident that RACNet significantly outperforms the top-performing methods in both competitions. On COV19D-ICCV2021, RACNet achieves a macro F1 score of 93.83%, surpassing FDVTS\_COVID by 3.4%, SenticLab.UAIC by 3.77%, and ACVLab by 5.09%. Similarly, on COV19D-ECCV2022, RACNet achieves a macro F1 score of 95.06%, outperforming MDAP by 7.19%, FDVTS by 5.95%, and ACVLab by 5.95%.

These results highlight RACNet’s effectiveness in COVID-19 detection using segmented CT scans, demonstrating its superior performance over state-of-the-art methods in competitive benchmarks.

### 3.6.3 Experiments on CC-CCII, MosMedData and CT-image Databases

In this section, we evaluate the performance of RACNet on three different databases: CC-CCII, MosMedData, and CT-image Database. We compare its performance with state-of-the-art methods and analyze its effectiveness in various classification tasks.

### CC-CCII Database

Table 3.5 illustrates the performance of RACNet for 3-class classification (novel coronavirus pneumonia, common pneumonia, and Normal) on the CC-CCII database using metrics such as accuracy, precision, sensitivity, and F1 score. Additionally, it compares RACNet with other state-of-the-art methods reported in Section ??.

RACNet significantly outperforms all methods in terms of all metrics. Specifically, RACNet achieves a F1 score that is 3.28% higher and accuracy that is 4.33% higher than EMARS-APS, the best-performing method. Compared to other state-of-the-art methods, RACNet achieves improvements ranging from 4.74% to 9.11% in F1 score and from 5.25% to 7.7% in accuracy. Notably, RACNet benefits from pre-training on COV19-CT-DB followed by fine-tuning on CC-CCII, leveraging feature priors from COV19-CT-DB for enhanced performance. Even when trained from scratch, RACNet maintains superior performance, outperforming EMARS-APS by 1.28% in F1 score and 2.39% in accuracy, and outperforming other methods by 2.76% to 7.11% in F1 score and 3.31% to 5.84% in accuracy.

Table 3.5 also includes a comparison of model sizes in MB between RACNet and the state-of-the-art methods. RACNet exhibits significantly lower model size compared to most methods, emphasizing its computational efficiency. The computational complexity of RACNet is approximately 112 BN FLOPs with about 4.4 million parameters.

### MosMedData Database

Table 3.5 further presents the performance of RACNet for COVID-19 versus non-COVID-19 diagnosis on the MosMedData database. Similar to CC-CCII, RACNet outperforms all state-of-the-art methods in terms of various metrics. Pre-trained RACNet on COV19-CT-DB achieves a F1 score that is 1.73% higher and accuracy that is 1.79% higher than EMARS-APS, and outperforms other methods by 5.65% to 5.84% in F1 score and 7.58% to

9.83% in accuracy. Even when trained from scratch, RACNet maintains superiority, surpassing EMARS-APS by 1.02% in F1 score and 1.05% in accuracy, and outperforming other methods by 4.94% to 5.13% in F1 score and 6.84% to 9.09% in accuracy.

Table 3.5 also shows the model size comparison, highlighting RACNet’s compact design compared to other methods.

### **CT-image Database**

Finally, Table 3.5 compares the performance of RACNet for COVID-19 versus non-COVID-19 diagnosis on the CT-image Database. RACNet surpasses VGG19 and ResNet50 by 10.47% and 19.95% in accuracy, respectively, despite having a significantly lighter model size.

In summary, the comprehensive comparison across these databases demonstrates RACNet’s superior performance in terms of classification metrics and efficiency in model size. RACNet proves to be a computationally efficient approach for COVID-19 detection across diverse datasets.

**Table 3.5:** Performance comparison between RACNet and the state-of-the-art on the test set of CC-CCII, MosMedData, and CT-image Database; Acc stands for Accuracy metric

| Dataset     | Method              | Size (MB) | Acc          | Precision    | Sensitivity  | F1           |
|-------------|---------------------|-----------|--------------|--------------|--------------|--------------|
| CC-CCII     | MC3_18 [133]        | 43.84     | 86.16        | 87.11        | 82.78        | 84.89        |
|             | Densenet3D121 [134] | 43.06     | 87.02        | 88.97        | 82.78        | 85.76        |
|             | COVID-AL [135]      | -         | 86.60        | -            | -            | -            |
|             | VGG-Ensemble [136]  | -         | 88.12        | 84.04        | 89.19        | 86.54        |
|             | MNas3DNet [137]     | 22.91     | 87.14        | 84.44        | 86.09        | 87.25        |
|             | CovidNet3D [122]    | 53.26     | 88.69        | 90.48        | 88.08        | 89.26        |
|             | EMARS-APS [138]     | 3.38      | 89.61        | 91.48        | 89.97        | 90.72        |
|             | <b>RACNet</b>       | 8.60      | <b>93.94</b> | <b>93.69</b> | <b>94.30</b> | <b>94.00</b> |
| MosMedData  | MC3_18 [133]        | 43.84     | 80.04        | 79.43        | 98.43        | 87.92        |
|             | Densenet3D121 [134] | 43.06     | 79.55        | 84.23        | 92.16        | 88.01        |
|             | DeCoVNet [139]      | -         | 82.43        | -            | -            | -            |
|             | CovidNet3D [122]    | 60.39     | 82.29        | 79.50        | 98.82        | 88.11        |
|             | EMARS-APS [138]     | 10.69     | 88.09        | 93.52        | 90.59        | 92.03        |
|             | <b>RACNet</b>       | 8.60      | <b>89.87</b> | <b>94.69</b> | <b>92.85</b> | <b>93.76</b> |
| CT-image DB | ResNet50 [140]      | 98.0      | 76.32        | -            | -            | -            |
|             | VGG19 [140]         | 549.0     | 84.80        | -            | -            | -            |
|             | <b>RACNet</b>       | 8.60      | <b>95.27</b> | <b>93.15</b> | <b>97.14</b> | <b>95.10</b> |

### 3.6.4 Anchor Set Creation

In this section, we detail the process of latent variable extraction and anchor set generation during the training of RACNet using the COV19-CT-DB database. This procedure was designed to enhance the interpretability and diagnostic capability of RACNet in identifying COVID-19 and non-COVID-19 cases based on chest CT scans.

The latent variable extraction and anchor set generation process aims to create representative vectors in a 128-dimensional space, encapsulating distinct patterns observed in the CT scans. These anchors serve as reference points that help classify new scans based on their similarity to known patterns. Specifically, we derived a total of 11 anchors through this process.

Each anchor vector is associated with a cluster center, capturing specific features and patterns present in the dataset.

Out of these 11 anchors, 7 were identified to correspond to COVID-19 cases, characterized by various degrees of pulmonary involvement indicative of COVID-19 pneumonia. The remaining 4 anchors were attributed to non-COVID-19 cases, encompassing different pulmonary conditions and normal lung scans.

To quantify the distribution of cases within each cluster, Table 3.6 provides a comprehensive overview. This table enumerates the number of CT scans assigned to each cluster along with their classification into COVID-19 or non-COVID-19 categories. Additionally, the severity of COVID-19 within each cluster is rated on a scale from 1 to 4, with higher scores indicating more severe manifestations of the disease. Table 3.7 elaborates on these severity categories, offering detailed descriptions for better clinical interpretation [123].

The centers of these 11 clusters collectively form the anchor set used in COV19-CT-DB. Each anchor represents a distinct pattern or anomaly observed in the chest CT scans, enabling RACNet to classify new scans based on their proximity to these predefined patterns. This approach enhances the diagnostic process by providing a structured framework for interpreting scan results and identifying key indicators of COVID-19 or other pulmonary conditions.

The utilization of these anchors in the classification of the COV19-CT-DB test set demonstrated robust performance, closely aligning with the original RACNet’s classification accuracy. This validation underscores the efficacy of the anchor-based approach in enhancing diagnostic capabilities and providing transparent decision-making in clinical settings.

For validation, we used this anchor set to classify the COV19-CT-DB test set. In particular, we fed each 3-D CT scan in the test set of the RACNet architecture; we extracted the corresponding dense layer neuron outputs; we computed their euclidean distance from each anchor. Then they were



**Table 3.6:** Number of CT Scans per cluster, cluster category & Severity category in COV19-CT-DB

| Cluster ID | Number of CT Scans | Category     | Severity Category |
|------------|--------------------|--------------|-------------------|
| 0          | 231                | COVID-19     | 3                 |
| 1          | 360                | COVID-19     | 2                 |
| 2          | 344                | COVID-19     | 4                 |
| 3          | 106                | COVID-19     | 1                 |
| 4          | 195                | COVID-19     | 4                 |
| 5          | 156                | COVID-19     | 3                 |
| 6          | 242                | COVID-19     | 4                 |
| 7          | 107                | non COVID-19 | 1                 |
| 8          | 586                | non COVID-19 | 1                 |
| 9          | 557                | non COVID-19 | 1                 |
| 10         | 322                | non COVID-19 | 1                 |

**Table 3.7:** Description of the Severity Categories

| Category | Severity | Description  |
|----------|----------|--|
| 1        | Mild     | Few or no Ground glass opacities. Pulmonary parenchymal involvement $\leq 25\%$ or absence |
| 2        | Moderate | Ground glass opacities. Pulmonary parenchymal involvement 25 – 50%                         |
| 3        | Severe   | Ground glass opacities. Pulmonary parenchymal involvement 50 – 75%                         |
| 4        | Critical | Ground glass opacities. Pulmonary parenchymal involvement $\geq 75\%$                      |

classified according to the label of their nearest cluster center. The obtained classification performance over the test dataset was similar to the original RACNet’s classification performance.

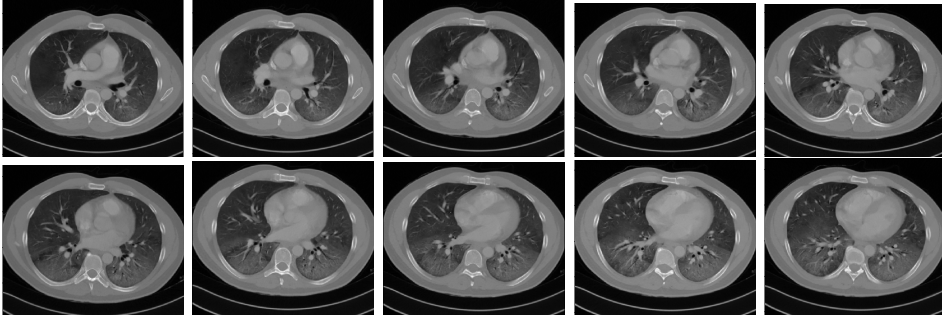
**Table 3.8:** Description of findings in each cluster center in COV19-CT-DB

| Cluster ID | Description   |
|------------|---|
| 0          | Bilateral shadows ground-glass that become more compact locally in lower lung lobes with an image of pneumonia due to COVID-19; severe category   |
| 1          | Bilateral shadows ground-glass as in pneumonia due to COVID-19; moderate category   |
| 2          | Minimal shadows ground-glass in left upper lung lobe. Severe thickening shadows, dense atelectasis of lower lung lobes. Minimal pleural fluid on the right. Possible microbial cause; critical category |
| 3          | Bilateral shadows ground-glass mainly in lower lung lobes as in pneumonia due to COVID-19 in rather mild condition; mild category   |
| 4          | Bilateral shadows ground-glass that occupy more than 75 % of the pulmonary parenchyma as in pneumonia COVID-19 of critical condition; critical category   |
| 5          | Bilateral shadows ground-glass that occupy about 50 % of the pulmonary parenchyma as in pneumonia COVID-19 of critical condition; severe category   |
| 6          | Bilateral shadows ground-glass that occupy more than 75 % of the pulmonary parenchyma as in pneumonia COVID-19 of critical condition; critical category   |
| 7          | Bilateral emphysematous lesions as in chronic obstructive pulmonary disease. Dense atelectasis in paravertebral right lung; mild category   |
| 8          | Normal CT scan; mild category   |
| 9          | Normal CT scan; mild category   |
| 10         | Normal CT scan; mild category   |

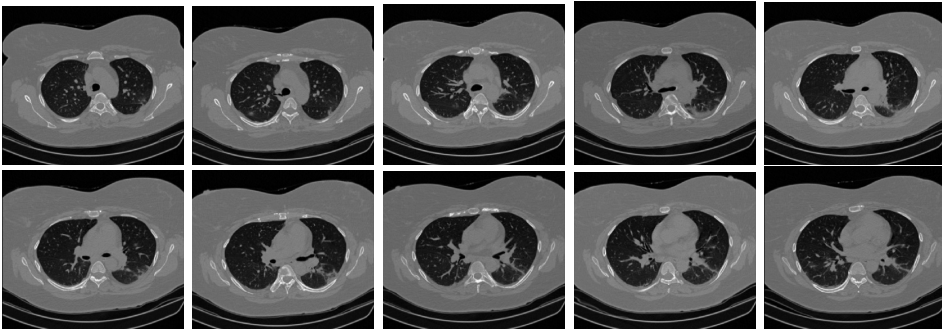
Moreover, our medical experts examined the 3-D scan inputs corresponding to the 11 cluster centres and produced justification for the respective diagnosis. Table 3.8 presents the findings detected in each cluster center.

Some examples of CT slices from the cluster centers are given below. Figure 3.7 shows 10 consecutive slices from COVID-19 cluster center 0. Medical experts have annotated it as 'bilateral ground glass regions that appear, especially in lower lung lobes'. Figure 3.8 shows 10 slices from COVID-19 cluster center 2. According to medical experts' annotation, this is con-

sistent with 'COVID-19 pneumonia bilateral thickening filtrates'. Figure 3.9, on the contrary, shows 10 slices from non COVID-19 cluster center 9.



**Figure 3.7:** Slices from cluster center 0 of COVID-19 category in COV19-CT-DB. Bilateral ground glass regions are seen especially in lower lung lobes.



**Figure 3.8:** Slices from COVID-19 cluster center 2 in COV19-CT-DB, which is consistent with COVID-19 pneumonia bilateral thickening filtrates.

The major advantage of the anchor set model is the insight that it introduces in the diagnosis process. In each new test case, the generated decision is accompanied by the information about the anchor to which this case was assigned through the above nearest neighbor classification procedure. As a result, the patient, or the doctor, can see which part of RACNet data-driven knowledge was used to make the specific diagnosis.

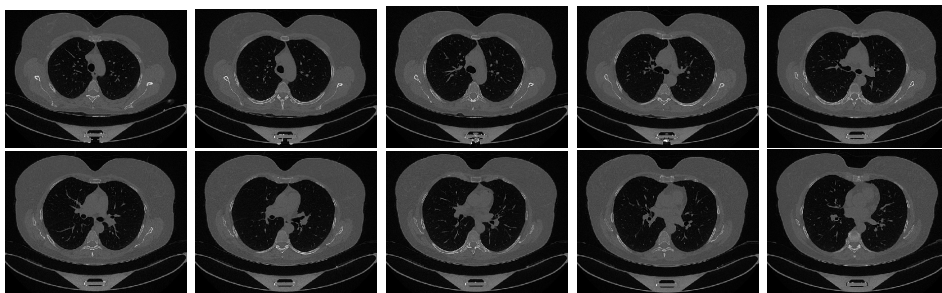


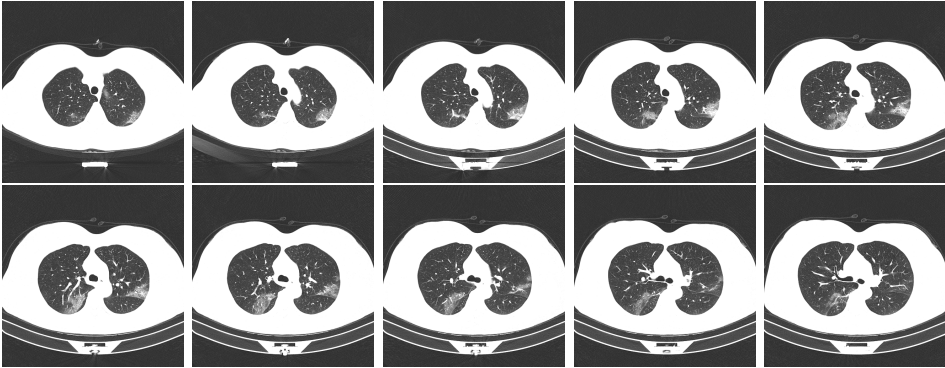
Figure 3.9: Slices from non COVID-19 cluster center 9 in COV19-CT-DB

### 3.6.5 Anchor Set Unification across Databases

In this section, we explore the utilization of RACNet trained on the COV19-CT-DB dataset to unify data-driven knowledge with similar databases, namely CC-CCII and CT-image Database. The objective is to leverage the anchor set generated from COV19-CT-DB to enhance classification capabilities across multiple datasets, thereby addressing issues such as 'catastrophic forgetting' in transfer learning and reducing computational overhead.

**Utilization of CC-CCII Database** Initially, we employed RACNet trained on COV19-CT-DB to extract 128-dimensional features for each CT scan in the CC-CCII database. These features served as inputs to train a neural network, denoted as  $NN^{(1)}$ , comprising three fully connected layers (64, 128, and 2 neurons respectively), aimed at predicting the COVID-19 status of CC-CCII data. Simultaneously, similar to the cluster extraction process described in Section 3.3.4, we extracted representations using RACNet and clustered them to generate 13 cluster centers that demonstrated optimal performance on the CC-CCII test partition. Figure 3.10 illustrates slices from one of the extracted COVID-19 cluster centers in CC-CCII.

Subsequently, we integrated the 11 cluster centers from COV19-CT-DB with the 13 cluster centers from CC-CCII to form a unified representation. This

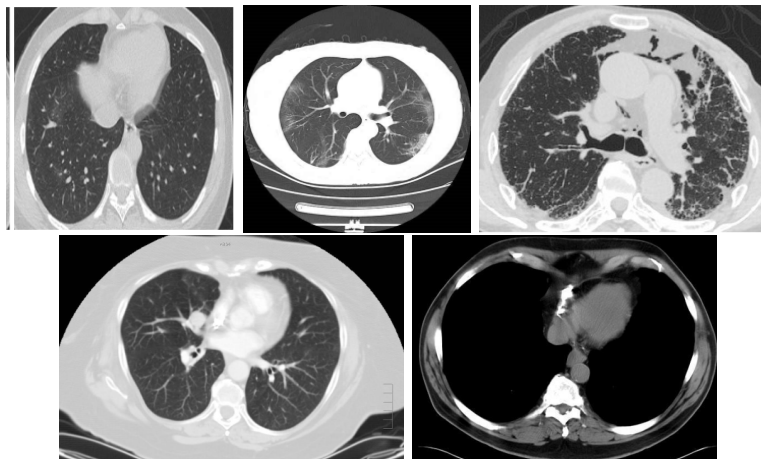


**Figure 3.10:** Slices of a new COVID-19 anchor in CC-CCII database, showing ground glass regions in the lungs.

unified representation leveraged the combined knowledge of RACNet and  $NN^{(1)}$ . To classify CT scans in both COV19-CT-DB and CC-CCII, we utilized the RACNet- $NN^{(1)}$  model, computing the nearest cluster center in the 128-dimensional space. The performance achieved was comparable to processing each database independently, demonstrating effective cross-database knowledge transfer without data exchange.

**Further Utilization of CT-image Database** To extend the unified representation approach, we incorporated the CT-image Database [124]. Using a similar methodology, we derived an additional set of 5 clusters from the CT-image Database, bringing the total number of cluster centers to 24 (11 from COV19-CT-DB, 13 from CC-CCII, and 5 from CT-image Database). We integrated these clusters into the unified representation alongside RACNet,  $NN^{(1)}$ , and a new neural network  $NN^{(2)}$  structured identically to  $NN^{(1)}$ . Figure 3.11 showcases the derived cluster centers from the CT-image Database, with two corresponding to COVID-19 and three to non-COVID-19 categories.

By merging these 24 cluster centers with RACNet,  $NN^{(1)}$ , and  $NN^{(2)}$ , the unified representation demonstrated robust performance across all three databases, akin to individual database processing. This approach under-



**Figure 3.11:** The 5 derived cluster centers in the CT-image Database; top three correspond to non-COVID-19 and bottom two to COVID-19 categories.

scores the efficacy of leveraging anchor-based representations and neural networks for seamless integration and classification across heterogeneous datasets, without compromising data privacy.

### 3.7 Deployment

The exponential growth in medical imaging data has underscored the need for efficient and timely diagnostic tools. Machine learning (ML) techniques have emerged as pivotal in augmenting clinical decision-making processes, particularly in computer-aided detection (CAD) systems for medical image analysis. These systems are essential in the context of COVID-19, playing a crucial role in identifying radiographic patterns indicative of infection. This aids healthcare professionals in making swift and accurate diagnoses. The integration of Artificial Neural Networks (ANNs), Machine Learning (ML), and Deep Learning (DL) models within CAD frameworks has demonstrated significant success in analyzing complex medical datasets [118, 141].

This thesis introduces a novel application for computer-aided diagnosis utilizing a microservices architecture, centered around our proposed state-of-the-art deep learning model known as RACNet. Key attributes of this system include its effectiveness, robust data anonymization techniques, fairness in decision-making, and enhanced explainability of AI-assisted diagnoses. The development and evaluation of this system leverage a comprehensive dataset, COV19-CT-DB [120,142], comprising 7,756 annotated 3D chest CT scans sourced from diverse medical institutions.

The exponential rise in the volume of medical imaging data, driven by advancements in imaging technology and the increasing reliance on imaging for diagnostic purposes, necessitates the development of efficient diagnostic tools. The application of machine learning (ML) techniques has been pivotal in this domain, significantly enhancing the capabilities of computer-aided detection (CAD) systems in medical image analysis. In the specific context of the COVID-19 pandemic, these CAD systems have become indispensable in identifying radiographic patterns indicative of COVID-19 infection, thereby aiding healthcare professionals in making rapid and accurate diagnoses.

The integration of Artificial Neural Networks (ANNs), Machine Learning (ML), and Deep Learning (DL) models within CAD frameworks has shown considerable promise. These advanced techniques have been particularly successful in analyzing complex medical datasets, enabling the detection of subtle patterns and anomalies that might be missed by human observers [118,141]. This thesis presents a novel application for computer-aided diagnosis, leveraging a microservices architecture and our state-of-the-art deep learning model, RACNet.

The development and evaluation of this system are based on the COV19-CT-DB dataset [120,142], which comprises 7,756 annotated 3D chest CT scans from various medical institutions. This extensive dataset allows for rigorous testing and validation of RACNet's capabilities.

Our deployment strategy for this AI application adopts a microservices architecture, a contemporary approach that offers significant benefits in terms

of scalability and security [143]. This architecture allows us to segregate critical operations, such as handling sensitive data, to edge environments (e.g., local doctors' stations), while offloading computationally intensive tasks to High-Performance Computing (HPC) cloud platforms. By doing so, we optimize resource utilization and ensure secure and efficient data processing.

The microservices architecture facilitates seamless data flow and automated communication processes, addressing the critical security concerns inherent in handling medical data. By distributing the workload across different environments, we achieve a robust and resilient system capable of operating effectively in various healthcare settings.

In summary, this thesis contributes to the field of medical imaging and AI by presenting:

1. A robust system architecture tailored for scalable and secure deployment of AI applications across heterogeneous computational environments.
2. An AI-enabled system capable of alleviating healthcare burdens in outbreak scenarios by prioritizing disease cases and improving diagnostic accuracy.

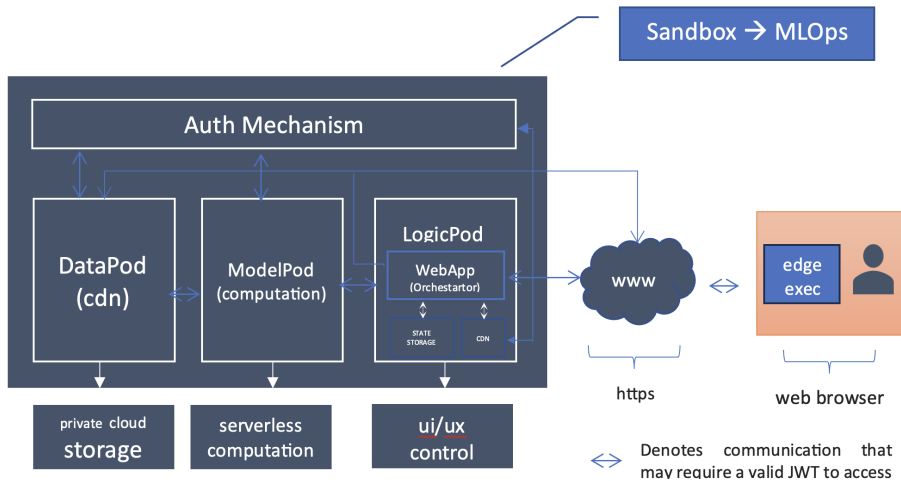
The proposed system exemplifies the potential of combining cutting-edge AI techniques with modern architectural frameworks to enhance healthcare delivery, particularly in times of crisis such as the COVID-19 pandemic.

### **3.7.1 MLOps Orchestration**

This section presents the MLOps orchestration of the RACNet model facilitated by MLPod™, a proprietary platform developed by AIandMe SMPC. MLOps represents a comprehensive approach to managing and deploying



ML applications, underpinned by a microservices-based architecture comprising four fundamental services: authentication mechanism, data hosting and sharing, model hosting and execution, and UI/UX. Figure 3.12 illustrates the overall architecture and information flow within MLPod™.



**Figure 3.12:** Architecture diagram illustrating the MLPod™ framework, depicting the flow of information between its constituent modules. Each module serves as an independent service, with rows indicating the direction of data flow. All communications are secured via encryption, highlighting MLPod™ as a comprehensive MLOps sandbox environment.

Each subsystem within MLPod™, termed Pods, is implemented as a separate Docker deployment, enhancing modularity and scalability. The logic microservice (UI/UX) incorporates a service discovery mechanism pre-configured with namespaces essential for deploying the RACNet model. Detailed functionalities of each MLPod™ module are outlined as follows:

**Auth Mechanism:** This subsystem is responsible for managing all authorization operations. It issues and validates access tokens in accordance with OAuth 2.0 standards [144]. These tokens regulate access to hosted data, RACNet model executions, and the deployed Web application. They embed access permissions and resource allocation constraints, ensuring secure and

controlled access to sensitive data and computational resources.

**DataPod:** Acting as the centralized data repository for the ML application deployment, DataPod facilitates data sharing using access tokens. This module stores COVID-19 cluster information, along with representative images and metadata crucial for model training and validation. DataPod ensures that all data interactions are secure and that data integrity is maintained throughout the lifecycle of the ML application.

**ModelPod:** This module is tasked with executing DICOM anonymization and RACNet-based COVID-19 detection tasks. Authorized through access tokens, ModelPod supports both cloud and edge computing environments. It automatically adapts to edge execution requests by encrypting and validating models before dispatching them for local execution. This feature ensures that sensitive patient information (e.g., DICOM tags) remains protected within local environments, thereby complying with data privacy regulations.

**LogicPod:** The LogicPod orchestrates application logic, generating UI/UX components as functional Web applications for end-user interaction. Leveraging the Machine Learning Markup Language (ML2), an XML-based document, LogicPod interprets data inputs, model configurations, and ML pipelines to render ML2 scripts into operational Web applications. The module integrates service discovery functionalities, specifying model services and execution environments (cloud and/or edge). Serving as a gateway orchestrator, LogicPod manages information flow, facilitates model prediction and inference, and presents AI-driven diagnostic reports to end-users in a user-friendly format. Updates to model parameters are seamlessly integrated into LogicPod deployments, ensuring real-time availability without disrupting application functionalities.

The MLPod™ platform's microservices-based architecture ensures that each component operates independently yet cohesively within the overall system. This modularity enhances scalability, allowing the system to adapt to varying computational demands and deployment scenarios.

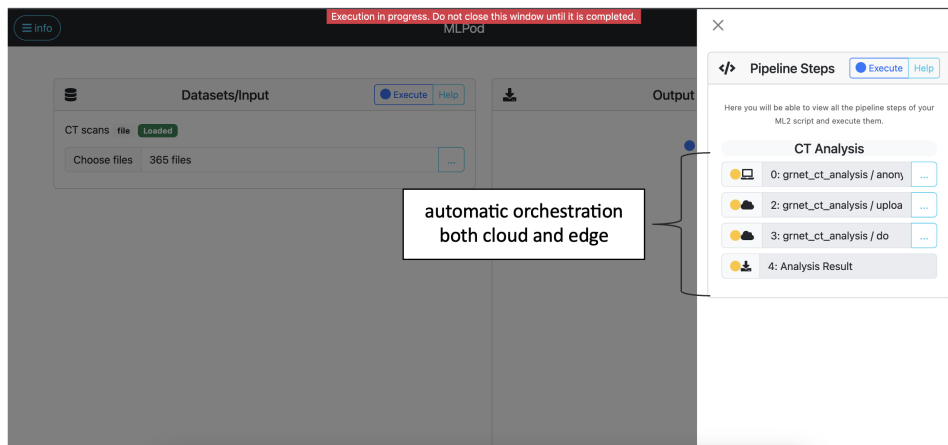
**Detailed Subsystem Functionalities:**

- **Auth Mechanism:** Ensures secure access management by issuing OAuth 2.0 compliant access tokens. These tokens manage permissions for accessing data, executing the RACNet model, and interacting with the Web application, embedding specific resource allocation constraints.
- **DataPod:** Acts as a secure repository for all data related to the ML application. It supports seamless data sharing via access tokens, ensuring that COVID-19 cluster data, representative images, and relevant metadata are accessible for model training and validation while maintaining data integrity and security.
- **ModelPod:** Manages the execution of DICOM anonymization and RACNet-based COVID-19 detection. It adapts to both cloud and edge environments by encrypting models for local execution, thus safeguarding sensitive patient data during edge deployments.
- **LogicPod:** Orchestrates the application logic, converting ML2 scripts into functional Web applications. It handles data inputs, model configurations, and ML pipeline specifications, integrating service discovery for seamless model deployment and execution across different environments.

Representative screenshots from a deployed RACNet-based COVID-19 detection application are depicted in Figures 3.14 and 3.15, illustrating the end-user (doctor) interaction and system workflow.

In conclusion, the MLPod™ platform offers a robust, scalable, and secure environment for deploying ML applications in healthcare. Its microservices architecture ensures modularity and adaptability, making it an ideal solution for managing complex workflows in AI-driven medical diagnostics.

Ethical considerations in AI-enabled medical diagnostics are paramount to



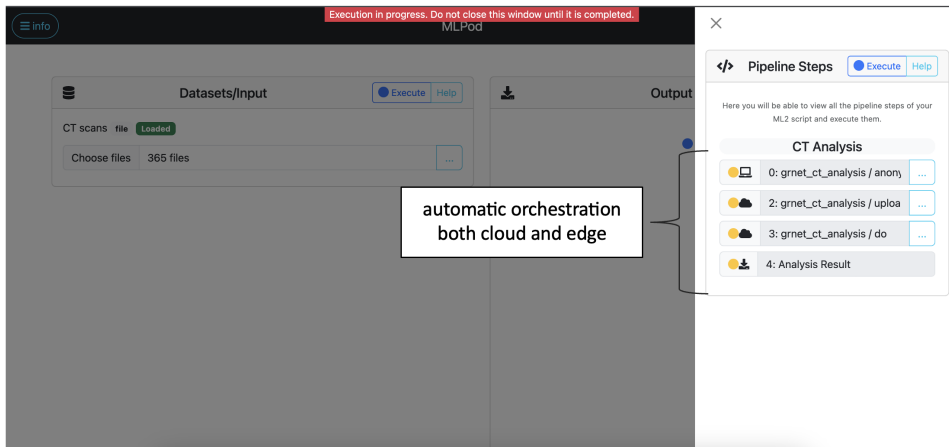
**Figure 3.13:** Screenshot of the RACNet-based COVID-19 detection application showing the initial data input interface for end-users.

ensuring patient confidentiality and regulatory compliance. This thesis adheres to established ethical standards in conducting a numerical simulation study and implementing a DL-based diagnostic system, for which ethical approval was deemed unnecessary. The utilization of the COV19-CT-DB dataset is fully acknowledged and detailed in Section 3.4.1.

### 3.8 Conclusion and Future Work

This thesis has presented a sophisticated system architecture designed for the rapid, secure, and scalable deployment of AI applications across heterogeneous computational environments. Central to this architecture is the RACNet model for COVID-19 detection, offering healthcare providers an intuitive, end-to-end interface for uploading DICOM images and receiving timely diagnostic outcomes accompanied by detailed explanations validated by RACNet’s decision-making process.

Future work will focus on leveraging user feedback to refine and enhance the RACNet model’s performance through iterative training and validation.



**Figure 3.14:** Screenshot depicting the execution of the RACNet-based COVID-19 detection application, highlighting the visualization of pipeline steps managed by LogicPod.

The continuous evolution of this MLOps journey aims to bolster diagnostic accuracy, adaptability to emerging clinical challenges, and overall user satisfaction within healthcare settings.

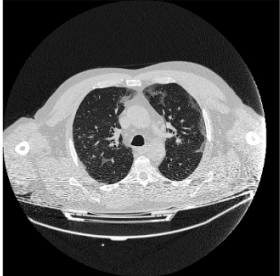
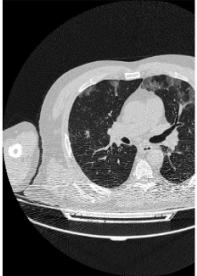
### Analysis Result

**Detection Result: Patient covid positive detected.**

**Severity Report:** Mild - Few or no Ground glass opacities. Pulmonary parenchymal involvement less than 25% or absense.

**Diagnosis Confidence:** 50%

**Classification Explainability**

|  |  |  |
|--|--|--|
|          |          |         |
| Slice 1: Bilateral shadows ground-glass as in pneumonia due to COVID-19; moderate category | Slice 2: Bilateral shadows ground-glass as in pneumonia due to COVID-19; moderate category | Slice 3: Bilateral shadows ground-glass as in pneumonia due to COVID-19; moderate category |

**HINT:** Click on each image above to zoom-in/zoom-out.

**Figure 3.15:** Illustration of the final diagnostic report generated by the RACNet model within the LogicPod framework. The report provides diagnostic outcomes, textual explanations, and representative medical images, aiding in clinical decision-making.

# Chapter 4

## Robustness in UAV Sense and Avoid

### 4.1 Introduction

Urban Advanced Air Mobility (AAM) is poised to revolutionize urban transportation, introducing a new paradigm for air traffic management and unmanned aerial vehicle (UAV) operations [145, 146]. However, the scalability of pilot availability remains a critical bottleneck in realizing the full potential of AAM, necessitating advancements in autonomous technologies [147].

Autonomous operation in AAM hinges on robust collision avoidance systems, crucial for navigating unmanned aircraft safely amidst a complex urban airspace [148, 149]. Current systems utilize onboard instruments like Automatic Dependent Surveillance-Broadcast (ADS-B) and Airborne Collision Avoidance System (ACAS) for cooperative traffic management, yet challenges persist in managing non-cooperative traffic effectively [150, 151]. Vision-based systems emerge as promising solutions due to their adaptability and robustness in diverse environmental conditions compared to other sensor modalities like radar and lidar [152–155].

This chapter addresses the critical need for out-of-distribution (OOD) robustness in UAV sense and avoid systems. While existing methods excel in controlled environments, they falter when confronted with novel or adversarial conditions, posing risks in real-world deployment scenarios [156]. Achiev-

ing robustness involves enhancing object detection accuracy and tracking reliability under varied conditions, from adverse weather to diverse aerial dynamics [157–160].

To advance the state-of-the-art, this work introduces NEFELI, a deep learning-based solution integrating vision-only detection, tracking and distance estimation modules on edge GPUs. NEFELI enhances object detection with a novel sliced inference technique, optimizing performance for real-time applications [161]. Additionally, it introduces a large-scale re-identification dataset and an innovative tracking module combining deep learning with Kalman filtering [162]. It also integrates a novel distance estimation model into the detection and tracking pipeline. Implemented on low SWaP edge GPUs, NEFELI demonstrates real-time efficacy and robustness against OOD data through extensive real-world experiments [163,164].

Moreover, this chapter presents AOT-C, a comprehensive benchmark dataset for evaluating the robustness of UAV detection algorithms under diverse real-world corruptions [163,165]. Through empirical studies, it assesses the vulnerabilities of state-of-the-art detection models to common corruptions, highlighting the need for adaptive algorithms in dynamic aerial environments.

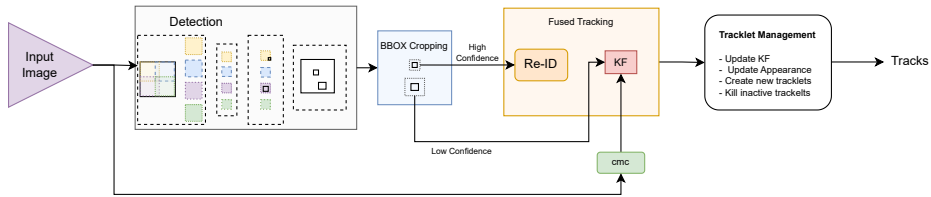
In summary, this chapter contributes novel methodologies and datasets to enhance the OOD robustness of UAV sense and avoid systems. By addressing the complexities of urban airspace management and non-cooperative traffic scenarios, it aims to advance the safety and reliability of autonomous UAV operations in urban settings.

## **4.2 NEFELI Architecture**

NEFELI is a system composed of multiple models, each specifically designed to effectively address the unique characteristics of detecting, tracking and estimating the distance of non-cooperative aircraft. To achieve this, the most suitable computer vision models were carefully selected and



enhanced with innovative algorithmic architectures, along with optimized training and inference strategies. The primary objective of this work is the creation of a novel deep-learning pipeline that not only surpasses the current state-of-the-art in detecting and tracking aerial vehicles but also is able to run in real-time on an edge device. The outputs of the NEFELI pipeline are expected to play a key role in informing collision avoidance algorithms by enabling automatic navigation of aircraft along collision-free paths.



**Figure 4.1:** Overview of NEFELI’s System Pipeline: Depicting the Detection and Tracking Modules.

NEFELI’s architecture is illustrated in Figure 4.6. The initial stage of NEFELI’s pipeline involves processing the input image through the detection module, which utilizes the sliced inference technique for accelerated inference on high-resolution images (details on the sliced inference technique are discussed in Section 4.2.1). Following the detection phase, the image sections containing aircraft bounding boxes are cropped, maintaining the bounding box at the center of the cropped image (BBOX Cropping). After the cropping step, the cropped images containing the detected aircraft undergo processing in the tracking component. In this stage, high-confidence detections are processed by the appearance Re-ID model while low-confidence detections are processed by the Kalman filter-based (KF) motion model. It is noted that the KF motion model takes into account camera motion compensation (CMC) effects to consider the impact of a moving camera. The tracking module updates the state of either the appearance model or the motion model, generating new tracklets to continue tracking an aircraft or eliminating inactive tracklets. The final output of the NEFELI pipeline consists of the tracked aircraft, including the bounding box of the detected aircraft and the corresponding track ID.

### **4.2.1 Air-to-Air Object Detection**

Selecting an appropriate algorithm for detecting aerial vehicles involves careful consideration of several key application characteristics. These characteristics include the high-speed motion of the target objects, their considerable distance from the camera source, and a low signal-to-noise ratio in the captured images. Additional challenges include occlusions and variations in appearance within the input images. Furthermore, it is imperative that the object detection and tracking models provide real-time inference to support automatic navigation decisions.

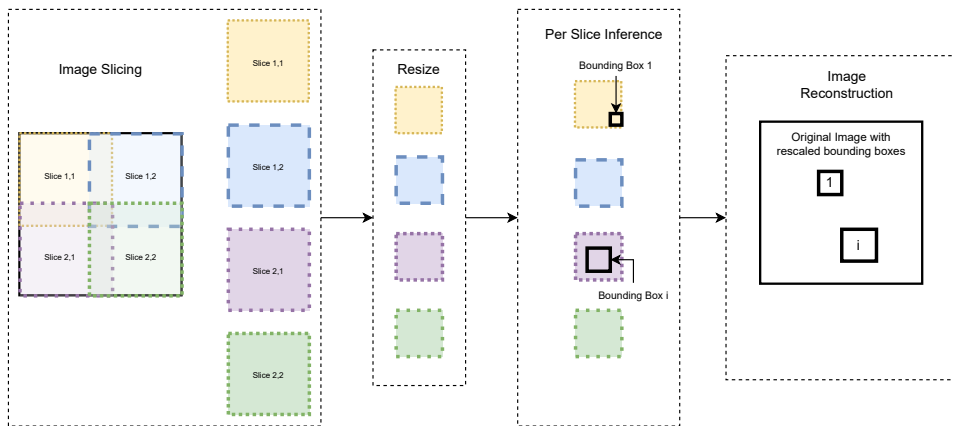
After a thorough evaluation of relevant studies in vision-based aircraft research, as discussed in Section 4.3.1, and the examination of various detectors in the public domain [166, 167], the YOLOv5 model from the You Only Look Once (YOLO) family has been identified as the most suitable choice. YOLOv5 optimally balances computational efficiency, precision, and recall. As the most mature version in the YOLO family, it offers enhanced accuracy and faster inference times compared to its predecessors. YOLOv5 introduces a streamlined architecture and employs anchor-based detection to improve its performance in addressing the specific challenges associated with aerial vehicle detection. Additionally, YOLOv5 incorporates advanced data augmentation techniques that are particularly beneficial for long-range detection tasks.

In terms of real-time inference, one-stage detectors like YOLOv5 generally outperform two-stage detectors. YOLOv5 achieves faster inference without compromising precision by utilizing focal loss during training to better handle challenging examples and to focus on difficult-to-detect objects.

Another notable strength of YOLOv5 lies in its capability to detect objects at various scales and orientations using anchor boxes and feature pyramids. The multi-scale approach and the use of anchor boxes of varying sizes enable the model to capture objects at different altitudes and orientations, including aerial objects.

However, one limitation of the YOLOv5 model is its reduced accuracy in detecting objects at long distances. Given that early detection of potential threats is crucial for aerial object detection in aircraft collision avoidance systems, the NEFELI system addresses this issue by enhancing the YOLOv5 model with an inference-time sliced inference step.

The proposed sliced inference method draws inspiration from related works by Akyon et al. [161] and Van Etten [168], and builds on top of YOLOv5. The architecture of the sliced inference technique is illustrated in Figure 4.2. In this technique, the input image is initially divided into smaller overlapping patches. These patches are then resized and independently processed through the detector. The outcomes from all patches are combined using the Non-max Suppression algorithm, which eliminates duplicate or highly overlapping bounding boxes. This merging process results in a more precise and concise set of detections, especially for distant objects. The final output of the sliced inference technique is the original high-resolution image containing appropriately re-scaled bounding boxes. An example of the slices of a high-resolution image is presented in Figure 4.3.



**Figure 4.2:** Sliced inference illustrated process (example with 4 slices).



**Figure 4.3:** An example of the slices of a high-resolution image.

#### **4.2.2 Synthetic Common Corruptions for Enhancing Robustness in Air-to-Air Object Detection**

A significant challenge in achieving fully autonomous flights lies in autonomous aircraft navigation, especially when dealing with non-cooperative traffic. The most effective strategy for managing such traffic involves processing monocular video feeds through deep learning models. This thesis advances the field of vision-based deep learning for aircraft detection and tracking by examining the effects of data corruption due to environmental and hardware conditions on these methods' effectiveness. Specifically, we developed seven types of common corruptions for camera inputs, simulating real-world flight conditions. By applying these corruptions to the Airborne Object Tracking (AOT) dataset, we created the first robustness benchmark dataset, named AOT-C, for air-to-air aerial object detection. The corruptions in this dataset span a wide array of challenging conditions, including adverse weather and sensor noise.

Historically, comprehensive datasets specifically designed for training deep learning algorithms in aerial object detection have been scarce. For example, Opromolla and Fasano [150] developed a dataset with a very limited number of images and a low resolution of 150x150 pixels, resulting in poor detection accuracy. More recently, Zheng et al. [157] introduced the DetFly dataset, which includes diverse backgrounds and consists of 13,271 images of a target micro UAV (DJI Mavic). Lee et al. [151] combined the DetFly dataset with their experimental dataset to enhance the performance of a YOLOv4-tiny model-based detector.

Several studies, including [162,169], leveraged advanced deep learning-based object detectors and trackers, using the AOT dataset [158] for training and evaluation. Introduced in 2021 as part of the Airborne Object Tracking Challenge hosted by Amazon Prime Air, the AOT dataset comprises approximately 5,000 flight sequences, totaling 164 hours of flight data with over 3.3 million labeled image frames. To our knowledge, the AOT dataset remains the largest and most comprehensive dataset for aerial object detection and tracking.

Numerous studies [165,170–172] have demonstrated that Deep Neural Networks (DNNs) are susceptible to common corruptions. For instance, [172] emphasized the necessity of detecting objects despite image distortions or adverse weather conditions for practical deep learning applications, such as autonomous driving. Corruption benchmarks were initially introduced in image recognition [173] and later extended to 3D object detection [163], semantic segmentation [174], pose estimation [175], and person re-identification [176]. Simulated imagery is also used for air-to-ground object detection [177,178].

However, many investigated corruptions are hypothetical and may not accurately represent real-world scenarios in autonomous UAV navigation. Additionally, in the context of air-to-air aerial detection, the objects are typically small. Therefore, artificially generated corruptions must be crafted to ensure the object’s visibility across all severity levels. Developing a comprehensive benchmark for evaluating the robustness of air-to-air aerial object detection under diverse real-world flying conditions remains a challenging task. To the

best of our knowledge, this work introduces the first robustness benchmark dataset for aerial object detection.

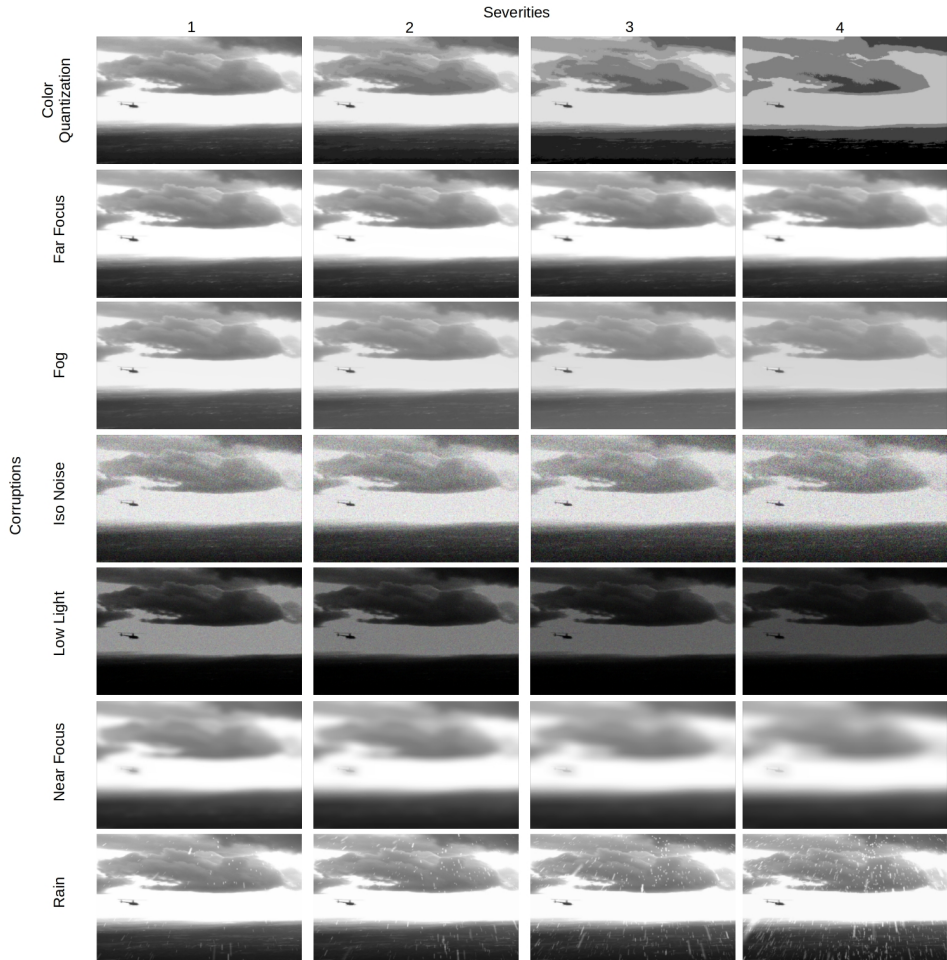
### **The Proposed Synthetic Dataset**

To conduct a comprehensive assessment of the corruption robustness of small object detection models (aerial object detection in our case), we establish a synthetic benchmark dataset using the widely used AOT dataset [158]. The selected corruptions are then applied to the test set of this dataset, resulting in the AOT-C benchmark. It is worth noting that while some corruptions may naturally occur in a few samples of the datasets, we apply synthetic corruptions uniformly across all data. This ensures a fair comparison of model robustness under different corruptions and streamlines the process of data filtering.

We illustrate seven corruption types in Figure 4.4 and classify them into three categories based on the typical presentations of common corruptions: weather, noise, and defocus. This dataset represents an initial endeavor, encompassing representative but not exhaustive corruptions. We encourage ongoing efforts to include a more diverse range of corruptions in future work. Brief introductions to each corruption pattern are provided below.

**Weather Corruptions.** Visual perception through cameras is susceptible to adverse weather conditions like rain and fog, where dense droplets of liquid or solid water can diminish the intensity of reflections and lower the signal-to-noise ratio (SNR) of received light. Moreover, floating droplets may produce false alarms and deceive sensors. These effects can significantly impact detectors in certain scenarios. To replicate three weather corruptions—rain, low-light (cloudy), and fog—we utilize simulators like [160] for rain and [171] for fog and low-light conditions. For simulating low-light scenarios, we decrease pixel intensities and use Poisson-Gaussian distributed noise, mimicking imaging conditions in low-light settings based on [171, 179].

**Sensor Noise Corruptions.** Noise corruptions arise due to constraints



**Figure 4.4:** Visualization of the seven corruption types for each severity level in our benchmark

inherent in camera sensors. ISO noise follows a Poisson-Gaussian distribution, characterized by consistent photon noise (represented by a Poisson distribution) and varying electronic noise (represented by a Gaussian distribution) [170, 171]. Furthermore, we incorporate color quantization as an additional corruption, decreasing the bit depth of the RGB image [171].

**Defocus Corruptions.** Blurring due to defocus in a moving camera video may occur when the camera lens fails to achieve a sharp focus on objects within the scene. Several factors can contribute to this, such as abrupt changes in the distance between the camera and objects, swift movements of the camera, or constraints in the camera’s autofocus system. In scenarios involving a moving camera, particularly at high speeds or in situations with frequent depth-of-field changes, achieving precise focus on objects becomes challenging. Defocus blur is commonly observed in such cases, resulting in a lack of sharpness and clarity in parts of the video where objects may appear blurry or out of focus.

### **Discussion on the Disparity Between Synthetic and Real-World Corruptions**

Corruptions in the real world can stem from a multitude of diverse sources. For example, an autonomous UAV might experience adverse weather conditions and encounter uncommon objects simultaneously, leading to more intricate corruptions. While it is impractical to list all potential real-world corruptions, we systematically categorize seven corruptions into four levels, creating a practical testbed for controlled robustness evaluation.

For weather-related corruptions, we utilize state-of-the-art simulation methods that closely approximate real data [160,170,171]. Although an inevitable gap exists, we validate that the model’s performance on synthetic weather aligns consistently with its performance on real data under adverse weather conditions by conducting real-world flight tests.

Each corruption type exhibits four severity levels, representing different intensities of manifestation. An example depicting four severity levels for each type of our synthesized corruptions is illustrated in Figure 4.4. These corruptions are implemented using functions, enabling seamless integration into the data loader for enhanced portability and storage efficiency.



### 4.2.3 Aerial Object Tracking

Following the detection component, the next module in the NEFELI pipeline is the tracking module. Tracking airborne objects is challenging due to factors such as high-speed motion, complex flight dynamics, occlusions, appearance variations, and sensor limitations. The primary difficulties in aircraft tracking involve compensating for motion effects caused by both the moving obstacle and the moving camera source, as well as the tracking model’s ability to effectively associate features of the detected object. This association is crucial for achieving successful and continuous tracking.

To address the camera motion compensation challenge, we adopt the global motion compensation (GMC) technique utilized in the OpenCV implementation [180] of the Video Stabilization module with an affine transformation [181].

In the remainder of this section, we present NEFELI’s tracking approach, which is based on the Strong-SORT tracker [182]. The primary innovations in NEFELI’s method include the introduction of an appearance model, which involves creating the first large-scale re-identification dataset of aerial objects to train the appearance model, and the fusion of a deep learning appearance feature model with a Kalman filter-based motion estimation technique in air-to-air aerial object tracking. These advancements will be analyzed in the following subsections in the aforementioned order.

#### Appearance (Re-identification) Model

For the appearance model, we employ an architecture named OSNet-EFDM, which builds on the Omni-Scale Feature Learning (OSNet) [183] CNN network and Exact Feature Distribution Matching (EFDM) [184] layers.

The OSNet network paradigm is followed, and its generalization ability is enhanced by adding layers that conduct feature distribution matching. Unlike the original model, which uses AdaIN [185] and assumes a Gaussian

prior (matching only mean and standard deviation), EFDM employs high-order statistics (beyond mean and standard deviation) to represent style information more accurately.

OSNet-EFDM is a model that can perform reliably across various scenarios, adapt to different datasets without extensive retraining, and provide insights into the learned features. The architecture of OSNet-EFDM excels at extracting informative representations that can effectively discriminate between visually similar objects, even in challenging aerial environments.

### **Fusion of Deep Appearance Feature Model and Kalman Filter-based Motion Estimation Technique**

A key innovation of this work is the fusion of a deep learning appearance model and a Kalman Filter-based motion model to create a tracking module tailored to the specific requirements and challenges of long-range detection and tracking of airborne objects. The proposed approach carefully considers the unique characteristics of this domain, making necessary adjustments and optimizations to ensure reliable and accurate tracking performance in demanding scenarios. By integrating motion and appearance information, NEFELI's model addresses the complexities of long-range aircraft tracking, paving the way for improved situational awareness and enhanced object detection and tracking capabilities.

In this work, we adopt an architecture that utilizes both high and low confidence detections, inspired by the paradigm developed by Zhang et al. [186]. High-confidence detections serve as strong candidates for initiating new tracks. When a high-confidence detection is made, a new track is created and associated with the corresponding object. High-confidence detections provide a reliable starting point for tracking and assist in establishing a strong initial association between objects and tracks. A key feature of the proposed model is that instead of discarding low-confidence detections of aerial vehicles, these are used to refine the state estimation of the corresponding tracks. By incorporating information from low-confidence detections [182, 186], the accuracy and robustness of the existing tracks are

improved.

Regarding the motion model, it is built upon the modification of the Kalman filter (Noise Scale Adaptive (NSA) Kalman algorithm), developed by Du et al. [187]. The conventional Kalman filter is susceptible to issues associated with low-quality detections [188] and overlooks information concerning the scales of detection noise. To address this challenge, the NSA Kalman algorithm introduces a formula for adaptively computing the noise covariance.

Within NEFELI’s tracker, the motion model is used for low-confidence detections, where the detected objects are either at a very long range, appearing as tiny areas in the image, or are partially occluded. In these cases, the motion model is more appropriate than the appearance model, which requires extracting semantic features of the object.

For the appearance model, only highly confident detections are considered, due to the susceptibility of appearance features to occlusion and blurring from objects at a very long range. As described by Zhang et al. [189], the exponential moving average (EMA) mechanism (Eq. 4.1) is utilized to update the appearance state  $e_i^k$  of the matched  $i$ -th tracklet at frame  $k$ .

$$e_k = \alpha e_i^{k-1} + (1 - \alpha) f_i^k \quad (4.1)$$

The appearance embedding of the current matched detection, denoted as  $f_i^k$ , is incorporated in Eq. 4.1 along with a momentum term  $\alpha = 0.9$ . To determine a match between the averaged tracklet appearance state  $e_i^k$  and the new detection embedding vector  $f_i^k$ , their cosine similarity is evaluated. To accomplish this, both the motion cost ( $d_{i,j}^{iou}$ ) and the appearance cost ( $d_{i,j}^{cos}$ ) are needed. The motion cost is the Intersection over Union (IoU) distance between the predicted bounding box of tracklet  $i$  and the detection bounding box  $j$ , while the appearance cost is the cosine distance between the average appearance descriptor  $i$  and the appearance descriptor of detection  $j$ .

#### **4.2.4 Monocular Distance Estimation**

This chapter presents a deep-learning framework that uses optical sensors to estimate the distance of non-cooperative aerial vehicles. Implementing this comprehensive sensing framework requires depth information, which is essential for enabling autonomous aerial vehicles to perceive and navigate around obstacles.

We propose a method for estimating the distance of a detected aerial object in real-time using input from a monocular camera. To train our deep learning component for depth estimation tasks, we utilize the Amazon Airborne Object Tracking (AOT) Dataset. Unlike previous approaches that integrate the depth estimation module into the object detector, our method formulates the problem as image-to-image translation. We employ a separate lightweight encoder-decoder network for efficient and robust depth estimation. The object detection module identifies and localizes obstacles, conveying this information to both the tracking module for monitoring obstacle movement and the depth estimation module for calculating distances. Our approach is evaluated on the AOT dataset, which is, to the best of our knowledge, the largest air-to-air airborne object dataset.

Concerns regarding mid-air collision (MAC) and near mid-air collision (NMAC) are significant in both manned and unmanned aircraft operations, especially in low-altitude airspace. Sense and avoid refers to an aircraft's capability to maintain a safe distance from and avoid collisions with other airborne traffic. Under visual flight rules, pilots mitigate NMAC/MAC threats by visually detecting and avoiding other aircraft to ensure safe separation. For medium to large airborne systems, active onboard collision avoidance systems such as the Traffic Alert and Collision Avoidance System or the Airborne Collision Avoidance System rely on transponders in cooperative aircraft. However, not all airborne threats can be tracked using transponders, presenting challenges for reliable operations in scenarios involving rogue drones, gliders, light aircraft, and inoperative transponders.

Ensuring aviation safety is paramount. Human vision acts as the final line

of defense against mid-air collisions, underscoring its critical role in aviation safety. To aid pilots in mitigating mid-air collision risks, machine vision can be employed to provide alerts regarding potential aircraft and objects in the airspace. Radar usage is often impractical due to the size, weight, and power (SWaP) limitations of Unmanned Aerial Systems (UASs). As a result, machine vision, utilizing CNN-based networks, has emerged as a promising avenue of research to address these challenges.

Machine vision is a widely explored area in onboard systems, enabling machines to perceive their surroundings. With the rapid advancement of computer vision, machine vision has emerged as a promising technology for identifying potential threats. Various approaches exist for object detection, including one-stage and multi-stage detection pipelines. Deep learning, in particular, has gained significant traction in machine vision for its capabilities in object detection, tracking, and depth estimation.

Cutting-edge approaches commonly employ Convolutional Neural Networks (CNNs) to extract features for predicting depth values per pixel, surpassing classical techniques by a wide margin. However, these methods rely on intricate and deep network architectures, leading to substantial computational overheads and impractical real-time execution without high-end GPUs. Consequently, deploying such methods on time-sensitive platforms like small drones becomes unfeasible. Conversely, a lightweight CNN-based encoder-decoder network can achieve real-time monocular depth estimation with a balance of accuracy and efficiency.

As indicated in the literature, real-time monocular depth estimation with accuracy and efficiency balance can be achieved using a lightweight encoder-decoder architecture. In this setting, the deep learning model is trained to translate the input image from the monocular camera to a depth mask, classifying every pixel into depth values.

### **Problem Formulation**

Our system, the NEFELI pipeline [169], consists of multiple models (illustrated in Fig. 4.6), each specifically designed to effectively address the

unique characteristics of detecting, tracking, and estimating the distance of airborne objects.

The proposed workflow begins with the input image being processed through the detection module. Subsequently, image sections containing aircraft bounding boxes are cropped to ensure the bounding box remains centered within the cropped image, a process known as BBOX Cropping. After cropping, the images containing the detected aircraft are processed in the tracking component. During this phase, high-confidence detections are managed by the appearance Re-ID model, while low-confidence detections are handled by the Kalman filter-based (KF) motion model. Notably, the KF motion model incorporates camera motion compensation (CMC) to account for the influence of camera movement. The tracking module updates the state of either the appearance model or the motion model, generating new tracklets to sustain the tracking of an aircraft or eliminate inactive tracklets.

In parallel with the tracking procedure, the images containing the detected aircraft are processed in the depth estimation model. Specifically, these images are passed through the encoder network, which extracts hierarchical features by progressively reducing the spatial dimensions while increasing the number of channels. These encoded features capture various levels of abstraction, including edges, textures, and object shapes. The encoded features are then fed into a decoder network, which upsamples the feature maps to the original resolution of the input image. The decoder produces a depth map as the output, where each pixel corresponds to the estimated distance of the corresponding pixel in the input image from the camera. The final outcome of the entire pipeline includes the tracked aircraft, complete with the bounding box of the detected aircraft, its associated track ID, and depth estimation information.

## **Dataset Configuration**

We conducted our training using the Airborne Object Tracking (AOT) [158] dataset. The AOT dataset is a comprehensive collection of approximately 5,000 flight sequences captured from aerial platforms such as drones, heli-

copters, and other air vehicles. These sequences encompass diverse environments, including urban and natural landscapes. Each sequence is accompanied by extensive annotations, including bounding boxes of tracked objects, distance information, geographic coordinates, and camera parameters.

For our study, we decomposed these sequences into individual frames and organized a dataset based on these images. We utilized the bounding box annotations and distance metadata to create ground truths for the depth estimation problem. Typically, depth estimation datasets [190] comprise image pairs: one image for training and another representing the depth map, where the values indicate the distance of objects from the camera.

The AOT dataset does not include such depth annotations, necessitating the use of provided information to construct depth maps. We leveraged the bounding box information and the distance of the object from the camera, determined via GPS technology. Given the coordinates of both the object and the camera, the distance is calculated as the difference between these coordinates, referring to the straight-line distance of the objects.

Using this information, we created depth maps where the values within the bounding box of the object are equal to the provided distance from the camera. For areas outside the bounding box, we assigned values corresponding to the maximum distances measured in the AOT dataset.

This approach enabled us to generate the necessary ground truths for training our encoder-decoder depth estimation model. While we made assumptions that may not fully align with real-world data, such as the bounding box strictly enclosing only the object of interest, the evaluation results on the AOT dataset indicate that the bias introduced by these assumptions is negligible.

### **4.3 Evaluation Methodology and Training Strategy**

To enable a rigorous evaluation of the NEFELI pipeline, each component is independently assessed and compared against corresponding state-of-the-art models. This section presents the datasets used for benchmarking, training, and evaluating each deep learning component of NEFELI’s pipeline, along with the metrics employed to assess their performance.

The datasets selected for the comparative analysis of the detector component were the MS COCO [191] dataset, recognized as a benchmark for evaluating detection, and the DetFly [157] dataset, which comprises high-resolution images ( $3840 \times 2160$  pixels) that include small UAVs. The detector’s comparative analysis is presented in Section 4.4.1.

Regarding the tracking module, only the appearance model relies on a deep learning algorithm and thus requires training. To train the appearance model, a re-identification dataset derived from the AOT dataset was created (airborne Re-ID dataset). We provide a comprehensive description of the methodology used to create the airborne Re-ID dataset, which uniquely incorporates a diverse range of aerial vehicles, including general aviation aircraft, helicopters, and UAVs. This dataset significantly contributes to enhancing the tracking module’s performance. The comparative analysis between NEFELI’s appearance model and alternative state-of-the-art models, evaluated on the airborne Re-ID dataset, is presented in Section 4.4.2.

Additionally, we evaluated the distance estimation module on the AOT dataset. This evaluation was crucial for validating the effectiveness and accuracy of our depth estimation approach in real-world scenarios.

NEFELI’s detection model (YOLOv5-l enhanced with the sliced inference technique) was trained on both the AOT and DetFly datasets, which are regarded as the most comprehensive datasets encompassing a diverse range of aircraft. For the detection model, a single detection class was considered, and all types of aircraft (aeroplanes, helicopters, and UAVs) were categorized under the umbrella of the “drone” class. Simultaneously, the



tracking appearance model was trained on the airborne re-identification dataset. In both instances, 80% of the data were utilized for training purposes, with the remaining 20% reserved for testing. The Graphics Processing Unit used for training and testing is the NVIDIA GeForce RTX 4090. Section 4.4.2 includes a comparative analysis between NEFELI’s tracker (comprising fused appearance and motion models) and other state-of-the-art trackers.

### 4.3.1 Air-to-Air Object Detection: Benchmark Datasets and Evaluation Metrics

To illustrate the key advantages of the YOLOv5-large model in terms of precision and speed compared to other detection models, a comparative analysis was conducted using two benchmark object detection datasets. The first dataset is the MS COCO [192], which is widely recognized as a benchmark in deep learning-based object detection. The second dataset is Det-Fly [157], comprising 13,271 high-resolution images of small UAVs. This dataset covers a diverse range of scenarios with varying viewing angles, background scenes, relative ranges, and lighting conditions.

The training strategy for YOLOv5 involves optimizing the model’s parameters using Stochastic Gradient Descent (SGD) with a specified learning rate schedule. The loss function for YOLOv5 combines multiple components: localization loss, confidence loss, and classification loss. The overall loss is a weighted sum of these components, balancing the importance of localization accuracy, prediction confidence, and correct classification. The loss function is defined as follows:

$$\text{Loss} = \lambda_{\text{coord}} \cdot (\text{Loc Loss}) + \lambda_{\text{conf}} \cdot (\text{Conf Loss}) + \lambda_{\text{cls}} \cdot (\text{Cls Loss}) \quad (4.2)$$

Here,  $\lambda_{\text{coord}}$ ,  $\lambda_{\text{conf}}$ , and  $\lambda_{\text{cls}}$  are weight parameters to adjust the contribution of Localization (Loc), Confidence (Conf), and Classification (Cls) loss respectively.

To evaluate the detector's performance, the following metrics were considered:

- Recall (Eq. 4.3), that measures the model's ability to correctly identify all the positive instances in the dataset, providing an indication of the model's ability to capture all relevant objects;

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (4.3)$$

where:

*TruePositives*(*TP*) number of correctly detected objects

*FalseNegatives*(*FN*) number of objects that were not detected by the model

- Precision (Eq. 4.4), that measures the accuracy of object detection by quantifying the proportion of correctly identified objects out of all the objects detected by the model;

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (4.4)$$

where:

*FalsePositives*(*FP*) number of incorrectly detected objects

- Average Precision (Eq. 4.5) is calculated as the average of maximum precision values at recall levels, normalized by the total number of positive instances.

$$\text{AP} = \frac{1}{n_{\text{pos}}} \sum_{r=1}^{n_{\text{pos}}} \max(\text{Precision}(r'), r' \geq r) \cdot \text{recall}(r) \quad (4.5)$$

where:

$n_{\text{pos}}$  total number of positive ground truth instances

$\text{Precision}(r)$  denotes the precision at the  $r$ -th recall level

$\text{Recall}(r)$  represents the recall at the  $r$ -th precision level

### 4.3.2 Aerial Object Tracking: Benchmark Dataset, Re-identification Dataset, and Evaluation Metrics

#### Benchmark Dataset

The Airborne Object Tracking (AOT) Dataset [158], released in 2021 as part of the Airborne Object Tracking Challenge by Amazon Prime Air, was used to evaluate the performance of NEFELI’s tracking model. This dataset includes approximately 5,000 flight sequences, each lasting 120 seconds and captured at a frequency of 10Hz, resulting in a total of 164 hours of flight data. It contains over 3.3 million labeled image frames featuring airborne objects. The images have a resolution of  $2448 \times 2048$  pixels and are grayscale with 8-bit depth. Annotations include bounding box and class labels, as well as range information for a subset of the dataset, with range values typically ranging from 200 to 2000 meters. The labeled objects vary in size, with areas ranging from 4 to 1000 square pixels. Approximately 55% of the planned airborne encounters have trajectories that could potentially lead to collisions. Regarding target positions, 80% are above the horizon, 1% are on the horizon, and 19% are below the horizon. The dataset captures various sky and visibility conditions, with 69% of the sequences having good visibility, 26% having medium visibility, and 5% depicting poor visibility conditions.

## Re-identification Dataset

To effectively track aerial objects, the appearance model of the tracking component is trained on a dataset specifically including general aviation aircraft, helicopters, and small UAVs. Datasets containing ground vehicles or people are inadequate because aircraft have distinct appearance features. Training on a suitable dataset enhances the model’s discriminative capabilities and improves the accuracy of matching during tracking.



**Figure 4.5:** Examples of substantial variances in viewpoints and scales of aerial objects from the airborne-Re-ID dataset used to train NEFELI’s appearance model

While person and vehicle re-identification have been extensively researched over numerous datasets, airborne ReID remains relatively unexplored. In the absence of an existing multi-view airborne dataset, the airborne-ReID dataset was created to train NEFELI’s appearance model (Figure 4.5). This empowers NEFELI’s tracking model to avoid identity switching, handle occlusions, and re-identify an aerial vehicle that temporarily moves out of the capture range.

Furthermore, incorporating a ReID dataset in the training of NEFELI’s tracking module opens the door to future multi-camera detection and avoidance systems that need to transfer tracking information (re-identification) from one camera to another.

To simulate the challenges encountered in Re-ID tasks, the airborne-Re-ID dataset combines the Temporally Near and Big-to-Small features introduced by Zhong et al. [193].

Temporally-Near evaluates performance over a short time span, where Re-ID modules within tracking frameworks must accurately identify the same airborne object in consecutive video frames. Temporal intervals of  $\frac{t}{5}$  and  $\frac{2t}{5}$  are chosen, ensuring relatively consistent target aerial vehicle sizes while accounting for variations in their viewpoints. This scenario replicates the challenge faced by a Re-ID module integrated within a tracking framework, where UAVs undergo viewpoint transformations within a limited range.

Big-to-Small assesses Re-ID performance across significant scale variations. Airborne detections are captured at intervals of  $\frac{t}{3}$ ,  $\frac{2t}{3}$ , and  $\frac{2t}{3}$  throughout the entire video, emulating the challenge of matching known airborne objects (with detailed visual information) with airborne objects detected from a considerable distance. This enables the identification of distant airborne objects and the evaluation of their potential threat level.

The dataset instances are sampled from the training set of the 4000 video sequences in the AOT dataset. Each instance is created by cropping airborne objects from single frames of these videos and resizing the airborne object images to  $90 \times 50$  (height  $\times$  width). Data augmentation techniques, such as random flipping, random cropping, and random erasing [194], as proposed by Organisciak et al. [193], are applied to enhance the dataset. Four images per identity are included for each setting, resulting in a total of 2509 airborne object identities and 10036 airborne object images. 80% of the identities are used for training, while the remaining ones are reserved for testing.

## **Appearance Re-identification Model Training Strategy and Evaluation Metric**

In deep learning re-identification tasks, the most widely used metric is the rank-1 metric (Eq. 4.6). This metric refers to the evaluation of the top-1 matching accuracy. It measures the performance of a model by determining if the correct match for a query image is ranked first among all the gallery images. Thus, the rank-1 metric quantifies the percentage of query images for which the model successfully identifies the correct match as the top-ranked result. It serves as a key indicator of the model’s ability to accurately match and identify individuals in re-identification tasks.

$$\text{Rank-1} = \frac{\text{Number of correct matches ranked 1}}{\text{Total number of queries}} \quad (4.6)$$

The rank-1 metric is calculated by dividing the number of correct matches that are ranked first (top-1) by the total number of queries. It represents the accuracy of correctly identifying individuals when the correct match is the top-ranked result.

The re-identification appearance model is trained on a labeled dataset containing pairs or triplets of images, where the goal is to learn feature representations that enable matching and identification of individuals across different camera views or time instances. The cross-entropy loss is used to train the re-identification model. Given a pair or triplet of images and their corresponding labels, the cross-entropy loss is computed based on Eq 4.7.

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \quad (4.7)$$

Here,  $N$  is the batch size,  $y_i$  is the ground truth label (1 if the pair or triplet is a match, 0 otherwise), and  $p_i$  is the predicted probability of a

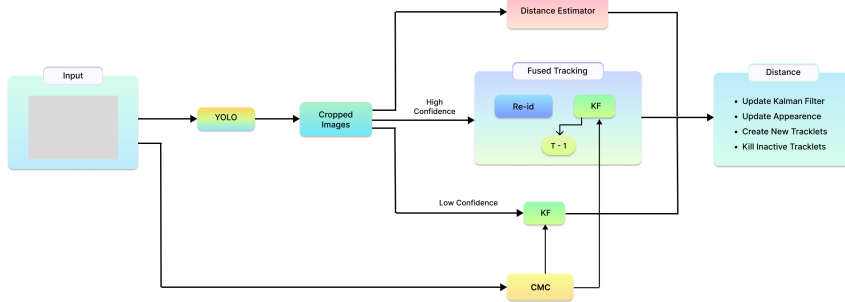


Figure 4.6: Nefeli's system pipeline

match provided by the model. The goal during training is to minimize this cross-entropy loss, encouraging the model to learn discriminative features.

### Tracking Evaluation Metrics

Major considerations for the selection of tracking evaluation metrics are aspects such as optimal tracking association and minimized identity switches, which are critical for autonomous aircraft navigation. Therefore, the following metrics have been considered in the following order of significance: Higher-Order Tracking Accuracy (HOTA) [195], Number of Identity Switches (IDsw) [196], Association Accuracy (AssA) [195], Detection Accuracy (DetA) [195], and False Positives Per Image (FPPI) [196] (this metric reflects the probability that a False Positive is detected in an image).

#### 4.3.3 Monocular Distance Estimation: Experimental Setup

Depth estimation [197] is typically framed as an image-to-image regression problem, where models predict a depth map corresponding in size to the

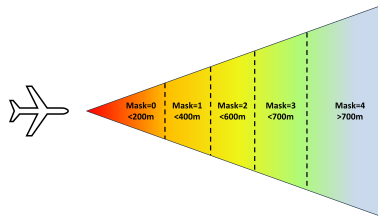
input image, with values indicating the distance of each object. Training involves minimizing the discrepancy between the predicted depth map and the ground truth.

Some studies [198] advocate for a classification approach to depth estimation, which can be particularly effective in UAV sense-and-avoid scenarios involving long distances. This method employs collision avoidance/safe separation thresholds [199] (see Fig. 4.7), framing the task as a multiclass classification problem with  $N$  depth increments ( $d_0, \dots, d_n$ ). Each increment represents a distinct class.

In this work, we adopt the classification framework. Based on distance information, we categorize the data into four different classes, with an additional class for background:

- Class 1: objects within 200 meters
- Class 2: objects within 400 meters
- Class 3: objects within 600 meters
- Class 4: objects within 700 meters
- Class 5: background objects beyond 700 meters

Using this classification scheme, we generated new ground truth *Masks* for training, resulting in  $W \times H$  arrays with values ranging from 0 to



**Figure 4.7:** Collision avoidance/safe separation thresholds



4.

$$Mask = \begin{cases} 0, & \text{if distance} < 200 \text{ meters} \\ 1, & \text{if distance} < 400 \text{ meters} \\ 2, & \text{if distance} < 600 \text{ meters} \\ 3, & \text{if distance} < 700 \text{ meters} \\ 4, & \text{if distance} > 700 \text{ meters} \end{cases} \quad W \times H \quad (4.8)$$

We trained an image-to-image model to produce outputs matching these mask labels. Although typical classification problems use cross-entropy loss, it is inefficient for our ordered data. Incorrect predictions in classes representing longer distances should incur greater penalties. To address this, we preprocessed input images by cropping around the center of the bounding box to minimize irrelevant information. A Gaussian filter was applied to the masks to smooth value divergences at the edges, normalizing the training data. We combined various loss functions such as the Structural Similarity Index (SSIM), edge-based losses, L1, and Berhu loss [200] to train a robust model capable of performing well under domain shifts [201].

- Edge loss retains object boundaries by penalizing false predictions at the object’s edge more severely.
- Structural Similarity Index (SSIM) measures image similarity based on luminance, contrast, and structural similarities.
- Berhu loss is a robust regression loss function designed to handle outliers while remaining sensitive to small errors.

We used a U-net convolutional neural network [202] for training. The U-net architecture includes encoder and decoder blocks. The encoder reduces spatial dimensions and increases feature map depth, while the decoder upsamples the feature maps to restore spatial resolution. Adam [203]

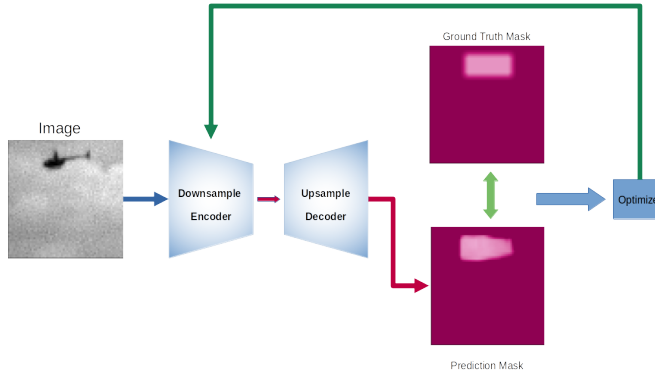
was used as the optimizer, with a weight decay of 0.0005 and an adaptive learning rate given by Eq. 4.9 to enhance training stability. To improve model generalization and reduce bias, L2 regularization [204] was utilized.

$$f(x) = \begin{cases} 1, & \text{if } x \geq \text{warmup iterations,} \\ \gamma * (1 - \alpha) + \alpha, & \text{if } x < \text{warmup iterations} \end{cases} \quad (4.9)$$

$$\text{where } \alpha = \frac{x}{\text{warmup iters}}, \gamma = 0.001,$$

$$\text{warmup iterations} = \min(1000, \text{length}(\text{dataset}) - 1)$$

The overall architecture of the depth estimation model is depicted in Fig. 4.8, where an image input is processed by the encoder-decoder model to produce a prediction mask closely matching the ground truth.



**Figure 4.8:** Pipeline of the proposed depth estimation model

## 4.4 Benchmark Evaluation

This section provides an in-depth analysis of NEFELI’s performance, detailing the experiments conducted and discussing their outcomes. The evalua-

tion process includes individual assessments of NEFELI’s detection model, appearance Re-ID tracking model, fused appearance and motion tracking models, and distance estimation module, followed by an evaluation of the entire NEFELI pipeline. A crucial aspect of this evaluation is the comparative analysis, in which each component of NEFELI, measured by the metrics presented in Section 4.3, is compared against other state-of-the-art models. The results of these analyses show that all of NEFELI’s components surpass the existing state-of-the-art models, and the comprehensive NEFELI pipeline also delivers superior results compared to the most advanced detection and tracking models available.

#### 4.4.1 Air-to-Air Object Detection

To benchmark air-to-air aerial object detection, we selected eight representative and diverse detectors: YOLOv5 [205], YOLOv8 [206], YOLOX [207], RetinaNet [208], Faster R-CNN [209], DiffusionDet [210], DETR [211], and CenterNet2 [212]. These detectors were chosen to cover various feature representations and proposal architectures.

For a fair comparison, each detector listed in Table 4.1 was trained on the clean training set of AOT following the training strategy specified in their respective papers. They were then evaluated using both the clean test set of AOT (first column of Table 4.1) and the corrupted test set of AOT (AOT-C) shown in the second column of Table 4.1.

The selected methods can be categorized into two groups based on their detection algorithms: one-stage networks (YOLOv5, YOLOv8, YOLOX, RetinaNet) and multi-stage networks (Faster R-CNN, DiffusionDet, DETR, CenterNet2). One-stage object detectors, like YOLO and RetinaNet, use a single unified network to simultaneously predict object bounding boxes and classify their content in one forward pass. In contrast, multi-stage detectors like Faster R-CNN use a region proposal network (RPN) to generate potential bounding box proposals which are then refined and classified by a subsequent network. DiffusionDet utilizes diffusion for detection by transitioning object boxes from ground-truth to random distribution during

training, and reversing this noise during inference. DETR uses the Transformer architecture, while CenterNet offers a probabilistic interpretation of the two-stage detection approach by using a strong first stage to estimate object likelihoods.

All detectors were executed based on open-source code available on GitHub. For YOLOv5, YOLOv8, and YOLOX detectors, we chose the large model type, as it is more suitable for small object detection [213]. Training and evaluation were performed on an NVIDIA RTX 4090 GPU with 24GB of memory. The batch size of each detector was optimized to maximize GPU memory usage. Default optimizers were employed, and parameters such as learning rate (LR), momentum, weight decay, and batch size were meticulously adjusted through extensive experimentation.

The standard evaluation was performed on the Aircraft, Helicopter, and small UAV categories, aggregated into one category called Drone. The evaluation metric was the Average Precision (AP) at an IoU threshold of 0.5. We denote model performance on the original validation set (AOT’s validation set) as  $AP_{clean}$ . For each corruption type  $c$  at each severity  $s$ , we used the same metric to measure model performance, denoted as  $AP_{c,s}$ . The corruption robustness of a model is calculated by averaging over all corruption types and severities as:

$$AP_{cor} = \frac{1}{C} \sum_{c \in C} \frac{1}{4} \sum_{s=1}^4 AP_{c,s} \quad (4.10)$$

where  $C$  is the set of corruptions in evaluation.

### Robustness Evaluation on AOT Dataset

Image corruptions reduce prediction accuracy. The robustness performance of the models, measured in APs, is shown in Table 4.1. Among all detectors, YOLOv5 demonstrates the highest robustness against corruptions ( $AP_{cor} = 53.5\%$ ), whereas RetinaNet performs the worst (20.0%). Among multi-stage networks, DiffusionDet is the most robust, achieving  $AP_{cor} = 35.7\%$ , while

**Table 4.1:** The benchmarking results of 8 object detectors on AOT and AOT-C in terms of Average Precision (AP), inference speed (fps), and model size (M)

| Object detector    | AP <sub>clean</sub> ↑ | AP <sub>cor</sub> ↑ | fps ↓      | Model Size ↓ |
|--------------------|-----------------------|---------------------|------------|--------------|
| YOLOv5 [205]       | 64.6                  | <b>53.5</b>         | 99         | 46.5         |
| YOLOv8 [206]       | 56.4                  | 41.2                | <b>110</b> | 43.7         |
| YOLOX [207]        | <b>69.3</b>           | 43.8                | 68         | 54.2         |
| RetinaNet [208]    | 35.7                  | 20.0                | 17         | <b>37.9</b>  |
| Faster R-CNN [209] | 52.9                  | 29.7                | 15         | 41.3         |
| DiffusionDet [210] | 63.8                  | 35.7                | 30         | 110.5        |
| DETR [211]         | 58.7                  | 26.1                | 27         | 41.2         |
| CenterNet2 [212]   | 66.2                  | 35.9                | 24         | 71.6         |

DETR, which relies on a Transformer encoder-decoder architecture, shows the lowest robustness (26.1%). For one-stage networks, all YOLO family models perform well (over 40%), in contrast to RetinaNet, which achieves only 20.0%. Although one-stage networks often trade detection performance for higher inference speed, YOLO models achieve better robustness on corrupted data and comparable performance on the original AOT evaluation set (clean) compared to multi-stage detectors like CenterNet2, DETR, and DiffusionDet. This suggests that YOLO models, particularly YOLOv5, could be a strong alternative for tasks requiring high computational efficiency and robustness to adverse conditions.

In summary, YOLOv5 and YOLOX demonstrate stable and superior performance compared to other detectors. Given their higher computational speed and lower parameter requirements (model size in millions of parameters), YOLOv5 and YOLOX are suitable choices for tasks with limited computational resources.

To further evaluate the performance of the algorithms, we present the results for each of the seven synthetically constructed corruptions separately in Table 4.2.

**Table 4.2:** The benchmarking results of 8 object detectors on AOT-C. We show the performance under each corruption and the overall corruption robustness  $AP_{cor}$  averaged over all corruption types

| Corruption            | YOLOv5      | YOLOv8 | YOLOX | RetinaNet | Faster R-CNN | DiffusionDet | DETR | CenterNet2 |      |
|-----------------------|-------------|--------|-------|-----------|--------------|--------------|------|------------|------|
| None ( $AP_{clean}$ ) | 64.6        | 56.4   | 69.3  | 35.7      | 52.9         | 63.8         | 58.7 | 66.2       |      |
| Weather               | fog         | 66.0   | 56.2  | 65.5      | 32.0         | 49.4         | 62.5 | 52.5       | 54.0 |
|                       | rain        | 64.2   | 53.8  | 64.3      | 32.4         | 49.9         | 61.1 | 50.6       | 55.2 |
|                       | low light   | 49.4   | 33.4  | 38.4      | 18.8         | 21.0         | 24.0 | 22.5       | 25.2 |
| Sensor noise          | color_quant | 49.8   | 41.2  | 42.3      | 19.8         | 35.0         | 38.9 | 10.8       | 35.6 |
|                       | iso noise   | 32.5   | 18.3  | 20.2      | 4.9          | 8.2          | 9.3  | 6.3        | 10.8 |
| Defocus               | far focus   | 58.3   | 50.2  | 51.6      | 25.0         | 38.4         | 48.3 | 38.7       | 44.4 |
|                       | near focus  | 44.5   | 36.3  | 37.9      | 16.1         | 24.9         | 32.8 | 22.3       | 32.8 |
| $AP_{cor}$            | 53.5        | 41.2   | 43.8  | 20.0      | 29.7         | 35.7         | 26.1 | 35.9       |      |

### Corruption Types that Affect Aerial Detection

How do different corruptions impact overall detector accuracy? As shown in Table 4.1, the average  $AP_{cor}$  is 22.7% lower than the  $AP_{clean}$ , indicating a significant decrease in detector accuracy when exposed to various corruption patterns. These findings highlight the urgent need to address the robustness challenges faced by aerial object detectors. Specifically, as indicated in Table 4.2, ISO noise, near focus, and color quantization corruptions cause the most significant AP loss. This results in a serious degradation in detection accuracy. Conversely, some corruption patterns (e.g., fog and rain) have a lesser impact on the detectors.

Our experimental study, detailed in Table 4.2, shows that adverse weather conditions do not significantly affect detector accuracy. In contrast, sensor noise, especially ISO noise, significantly degrades the object detection performance of all models.

### Network Attributes Affecting Robustness

As shown in Table 4.1, multi-stage detectors exhibit lower robustness against common corruptions compared to one-stage detectors, as evidenced by their lower  $AP_{cor}$ . One possible explanation is that corrupted data may impact proposal generation in the first stage (for both two-stage and multi-stage detectors), and poor-quality proposals can significantly affect the bounding box regression in the second stage (specifically for multi-stage detectors) [164]. Furthermore, as indicated in Table 4.2, multi-stage detectors appear

to be less susceptible to adverse weather conditions, displaying higher  $AP_{cor}$ , but are more vulnerable to sensor noise corruptions. Conversely, one-stage detectors, particularly those in the YOLO family, demonstrate more accurate results across all common corruptions.

#### 4.4.2 Tracking Appearance (Re-identification)

In this section, we evaluate the Re-ID model within NEFELI’s tracking component. Incorporating a deep learning Re-ID model into the tracking system is a notable innovation of this work, marking the first instance of a deep learning-based tracker for air-to-air aerial object tracking.

To assess NEFELI’s Re-ID model performance against other Re-ID models, we utilized the airborne Re-ID dataset introduced in Section 4.3.2. The state-of-the-art models were evaluated using the standard Rank-1 metric within the torchreid [214] framework.

The evaluation process involved dividing the test set into a query set and a gallery set, each containing 502 identities. For a given query image  $q$ , all gallery images  $g_i$  are ranked based on the likelihood that  $g_i$  matches  $q$ , indicating they depict the same airborne object. The rank- $r$  matching rate measures the percentage of query images with a correct gallery match within the top  $r$  ranks.

NEFELI’s Re-ID model is based on the OSNet (Omni-Scale Network) [183] combined with Exact Feature Distribution Matching (EFDM) [184]. OSNet is a CNN architecture specifically designed for object re-identification, aiming to enhance performance by fusing features from various scales within a residual convolutional block. Each stream within the block learns features at a different scale, and the final omni-scale features are dynamically generated by combining the outputs of all streams. Given the varying scales at which UAVs can be observed, OSNet is well-suited for addressing UAV re-identification challenges.

Table 4.3 compares various state-of-the-art Re-ID models on the airborne

Re-ID dataset. Notably, the OSNet + EFDM model [184] outperforms other approaches in terms of Rank-1. Despite being significantly lighter than ResNet-based models (the OSNet + EFDM model has fewer parameters), it achieved superior performance, making it the preferred choice for the appearance model in NEFELI’s tracking module.

**Table 4.3:** Results of state-of-the-art methods on our aerial object re-id dataset named airborne-ReID.

| Method                    | Params (M) ↓ | Rank-1 ↑    |
|---------------------------|--------------|-------------|
| MLFN [215]                | 32.5         | 69.1        |
| ResNet-50 [216]           | 20.5         | 69.5        |
| ResNet-18 [216]           | 11           | 71.1        |
| ResNet-34 [216]           | 13.6         | 73.3        |
| SE-ResNet-50 [217]        | 23           | 74.6        |
| OSNet [183]               | 2.2          | 77.1        |
| <b>OSNet + EFDM [184]</b> | <b>2.2</b>   | <b>78.7</b> |

#### 4.4.3 Fused Appearance and Motion Tracking

This section evaluates NEFELI’s tracking component, which is a significant innovation as it integrates appearance and motion information in a unified pipeline for the first time.

In the evaluation study, the same object detector (YOLOv5l + sliced inference) is used, except for the Baseline (first row), where the detector is YOLOv5l without sliced inference. The appearance model (OSNet + EFDM), trained on the airborne Re-ID dataset, is used in Bot-SORT-ReID, Deep OCSORT, and NEFELI’s (ours) fused tracking module. Other trackers rely solely on Kalman filter-based motion estimation. The detection confidence threshold is set at 0.60 for all tracking models. All models were also evaluated using the AOT dataset [158]. Key metrics for autonomous aircraft navigation, such as optimal tracking association and minimized ID switches, were prioritized. Thus, the following metrics were considered in order of importance: Higher-Order Tracking Accuracy (HOTA) [195], Number of Identity Switches (IDsw) [196], Association Ac-



curacy (AssA) [195], Detection Accuracy (DetA) [195], and False Positives Per Image (FPPI) [196].

The Higher-Order Tracking Accuracy (HOTA) protocol is the primary metric as it effectively combines detection, association, and localization accuracy into a single metric. Following HOTA, the IDsw metric is crucial for reliable trajectory prediction in autonomous aircraft navigation, as identity switches can disrupt trajectory planning and cause a loss of valuable time for essential maneuvers.

Additionally, AssA assesses association accuracy, DetA evaluates detection accuracy, and FPPI is relevant for autonomous aircraft navigation as it reflects the method’s overall false alarms.

Table 4.4 compares the tracking performance of our proposed pipeline to other state-of-the-art methods that either rely on motion estimation alone or combine motion and re-identification models. The Baseline method consists of a YOLOv5l (without sliced inference) model combined with StrongSORT that uses only motion estimation for tracking. All methods in the table follow ByteTrack’s [186] approach, which associates all bounding boxes, not just the high-score ones, resulting in more accurate object tracking.

The results in Table 4.4 clearly show that the fused (Re-ID + motion) methods outperform their motion-based counterparts in terms of HOTA, IDsw, and AssA, the primary metrics for aerial object tracking. The main differences between the compared tracking methods lie in the ways they compute the Kalman filter state vector (motion estimation) and the fusion of motion and appearance features.

NEFELI stands out with a HOTA score of 31.72, surpassing all other methods for aircraft tracking. NEFELI achieves top-ranking performance with only 52 identification switches (IDsw) and significantly outperforms other tracking models in the AssA metric, scoring 48.31.

The evaluation also reveals that the Deep-OCSORT model excels in the

**Table 4.4:** Comparison of proposed detection and tracking methods on AOT dataset. Our fused detection (Enhanced YOLOv5) and tracking method with appearance model (OSNet) trained on our airborne-ReID dataset has the best result in terms of the primary tracking metrics.

| Method   | HOTA $\uparrow$ | IDsw $\downarrow$ | AssA $\uparrow$ | DetA $\uparrow$ | FPPI $\downarrow$ |
|--|-----------------|-------------------|-----------------|-----------------|-------------------|
| Baseline   | 22.14           | 171               | 25.43           | 19.79           | 0.231             |
| Bytetrack [186]  | 23.34           | 251               | 25.56           | 22.21           | <b>0.187</b>      |
| Bot-SORT [181]   | 18.89           | 711               | 20.79           | 18.05           | 0.211             |
| Bot-SORT-ReID [181]  | 25.64           | 205               | 33.88           | 19.80           | 0.220             |
| OCSORT [218]   | 22.62           | 357               | 23.73           | 22.46           | 0.210             |
| Deep OCSORT [182]  | 27.37           | 155               | 20.51           | <b>36.83</b>    | 0.235             |
| Strong-SORT [182]  | 29.20           | 121               | 38.29           | 22.57           | 0.208             |
| <b>NEFELI (Ours)</b><br>(Enhanced YOLOv-5l + fused Strongsort) | <b>31.72</b>    | <b>52</b>         | <b>48.31</b>    | 21.01           | 0.245             |

DetA metric with a score of 36.83. Deep-OCSORT is designed to minimize tracking errors by incorporating a re-update stage to correct motion errors. Specifically, it adds a re-update stage, beyond the traditional "predict and update" stages, to prevent error accumulation by using virtual observations from historical time steps, optimizing the DetA metric.

Given NEFELI's aim to facilitate non-cooperative flight management and integrate into an aircraft's autonomous navigation system, priority has been placed on its superior performance in HOTA, IDsw, and AssA metrics. These metrics are crucial for addressing the challenges of aircraft tracking, ensuring optimal tracking associations, and minimizing ID switches for robust trajectory change decisions.

To support future work and fair comparisons, the exact flights from the AOT dataset used for evaluation, along with the evaluation kit based on the official MOT17 challenge [196] evaluation procedure, will be published.

#### 4.4.4 Monocular Distance estimation

A widely utilized and pragmatic metric for evaluating the efficacy of depth estimation is either the mean absolute error or the root mean square error. In consideration of the manner in which we have delineated our problem regular regression metrics might not be quite enlightening in discerning the performance of our model, for this reason, we proposed four metrics that could be used in the classification approach of the depth estimation task. These metrics provide a classification value that conveys precision in the categorization of each detected object in the correct classification bin. More specifically we utilized sliding a window across the area of interest (bounding box) to extract spatial information about the predicted distance of the object and employed methods like mean, max and min to combine the retrieved windows. Another, metric we applied was the rate of similarity between two pictures where the classification is considered truthful when the rate surpasses a given threshold. The metrics mentioned are expounded upon in the subsequent section below.

#### Evaluation Results on the AOT Dataset

In this section, we present and analyze the results of the conducted experiments for the different tested components. We also, analyze how each component affects the different losses on the overall performance of the introduced metrics. First, we trained our model applying only the edge loss function. Let  $Y(x, y)$  be the predicted map and  $I(x, y)$  be the input image, we calculate the edge loss by retrieving the gradients of the predicted map array in both the x and y-axis. After assessing these gradients we apply a smoothness factor with the help of the input image  $I(x, y)$  and obtain the loss value as the sum of the mean absolute values for the x and y axis respectively as shown in Eq.4.11

$$EL_{x,y} = e^{-\frac{1}{N} \sum_{i=0}^{i=N} |\nabla_{\hat{x}, \hat{y}} I(\hat{x}, \hat{y})|} \cdot \nabla_{x,y} Y(x, y) \quad (4.11)$$

The model trained with this loss achieves poor results depicted in Table 4.5. It's reasonable to anticipate this outcome, considering its intended function

to safeguard the structural intricacies and boundaries found within the input images and depth maps. Nevertheless, edge loss remains a pivotal element within the domain of depth estimation, as it contributes to the generation of depth maps characterized by enhanced smoothness and visual coherence, thereby portraying objects with greater clarity and contrast. Next, the model was trained using the L1 loss, as a standard loss function in regression tasks. From the regression metrics we can observe that models trained with this loss function are capable of producing satisfactory results. From Fig. 4.9, it could be deciphered that the L1 function trains the model to sufficiently pinpoint the area of interest but the outliers of the area are not adequately defined.

For the forthcoming experiment, we assessed the efficacy of the model trained to utilize the BerHu function [200]. This function incorporates the benefits of both the L1 loss and L2 so it is natural to expect a better performance which is something that can be observed in Table 4.5 as the BerHu loss [200] achieves better performance in both the Mean Absolute Error (MAE or L1) [219] and Root Mean Square Error (RMSE) [219] loss. To assess the viability of the BerHu loss as a substitute for the L1 loss function, we conducted an experiment wherein the model was trained using a combination of both loss functions. Subsequently, we evaluated the model’s performance. Our observations indicate that the model’s performance remained largely consistent, albeit marginally inferior compared to the model trained exclusively with the BerHu loss function. Lastly, all the above loss functions were combined using a weighted function described in Eq.???. Predictions for each of the four classification bins of interest using the multi-loss trained model are presented in Fig. 4.10

$$\begin{aligned}
 Loss &= W_{\text{edge loss}} \cdot EL(\hat{y}, y) \\
 &+ W_{\text{ssim}} \cdot SSIM(\hat{y}, y) \\
 &+ W_{L1} \cdot L1(\hat{y}, y) \\
 &+ W_{\text{berhu loss}} \cdot Berhu(\hat{y}, y)
 \end{aligned} \tag{4.12}$$

Except for the regression metrics used in evaluating the trained models, our assessment of accuracy relied on our proposed classification metrics. We

initiated our approach by devising a metric centered on a sliding window of dimensions  $5 \times 5$ . This window systematically traversed the designated area of interest (bounding box) within the predicted mask to gather spatial depth information at the specified distance. Subsequently, after traversing the window across the point of interest, we collected a series of  $m$  kernels, each with dimensions  $k \times k$ , to extract a singular value. For each kernel, we computed the mean of the  $k^2$  values, resulting in  $m$  values. Each value represented the estimated depth of its respective window. Following this step, we explored three distinct approaches to deriving the final prediction from the set of  $m$  values: utilizing the mean, minimum and maximum values. The resultant values ranged from 0 to 4, subsequently rounded to the nearest integer value (0, 1, 2, 3, or 4) to yield the final prediction.

Let  $M$  denote the predicted mask, and  $D(x, y)$  denote the depth information at coordinates  $(x, y)$ . For each window position  $(x, y)$  within the bounding box where  $(x, y)$  stand for the top left coordinates of the window, the depth  $D(x, y)$  value for each kernel is calculated by Eq.4.13:

$$D_{kernel}(x, y) = \frac{1}{k^2} \sum_{i=1}^k \sum_{j=1}^k M(x + i, y + j) \quad (4.13)$$

For the sliding windows, we utilized a stride value of 1 and applied padding to the predicted mask by reflecting its values. Let  $f$  be the function we applied to get the final prediction of depth (mean, min, max). The final

**Table 4.5:** Regression metrics for the different losses

| Model | Loss functions     | Regression Metrics |      |
|-------|--------------------|--------------------|------|
|       |                    | MAE                | RMSE |
| Unet  | Edge               | 3.39               | 3.55 |
|       | L1                 | 0.19               | 0.43 |
|       | Berhu              | 0.12               | 0.36 |
|       | L1/Berhu           | 0.13               | 0.37 |
|       | Edge/SSIM/L1/Berhu | 0.14               | 0.39 |

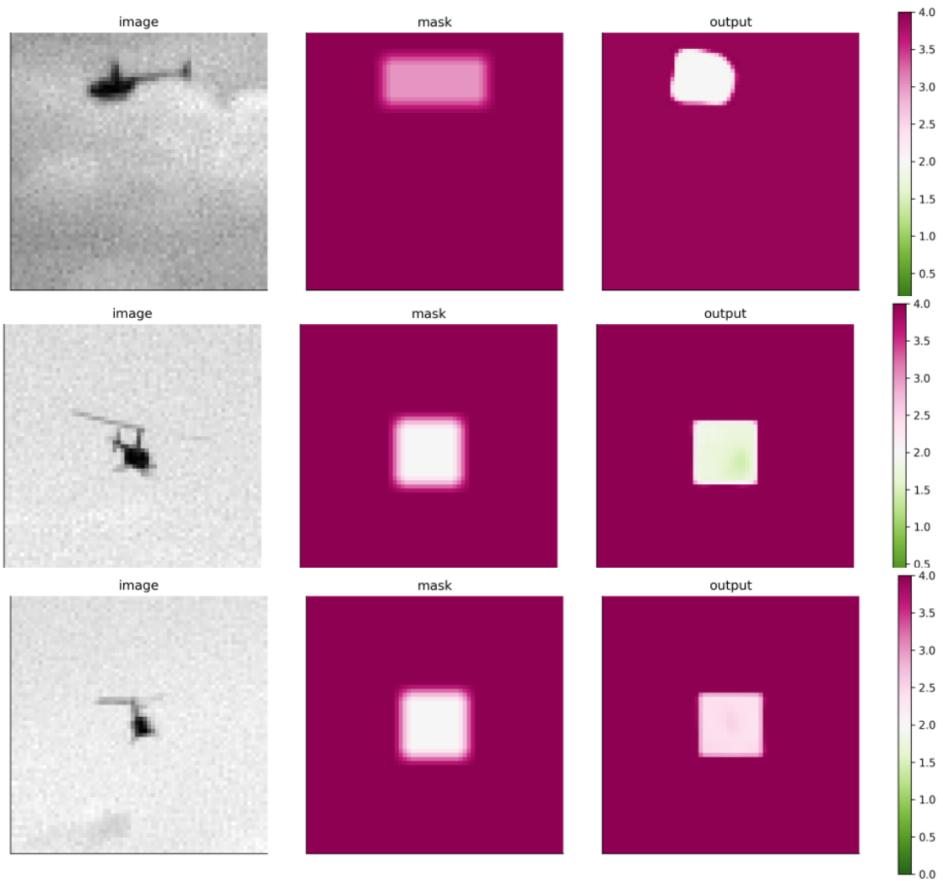
prediction can be described by the Eq.4.14

$$\text{class} = \text{round}\left(f\left(\sum_{i=1}^m D_i(x, y)\right)\right) \quad (4.14)$$

From Table 4.6 we can see that the worst-performing model is the one trained only with edge loss. This is expected as we observed the same in the regression metrics table (Table 4.5). Of all the three different functions tested in the Sliding Window metric, the worst-performing one is the max function. The decrease in performance can be attributed to the significance of outliers within the function.

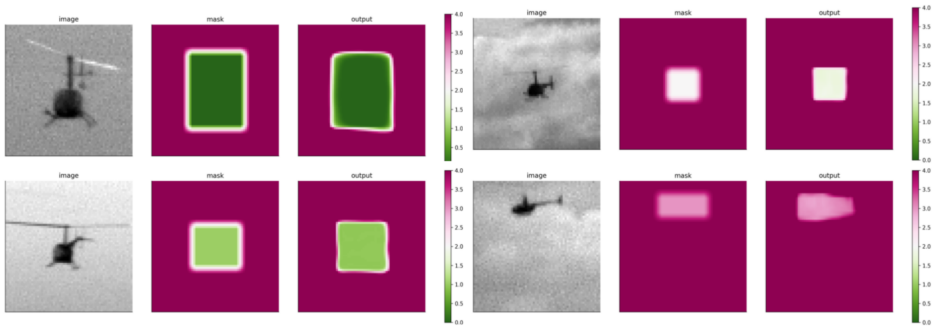
In particular, a thorough examination of the L1 prediction mask visualization depicted in Fig. 4.9 demonstrates that the bounding box boundaries predicted by the model using L1 loss function are inadequately defined. Since the background has larger values the max function chooses these as the correct categories diminishing the classification results. The accuracy achieved through the sliding window method utilizing the mean value demonstrates a notably superior performance in comparison to that attained through the max function. This outcome aligns with expectations, as the mean function effectively mitigates the influence of larger outlier values. Notably, among the three functions considered, the min function emerges as the most effective performer. Disregarding the larger values introduced by outliers, the minimum function results in the highest performance among the evaluated methods.

We additionally employed a metric, known as threshold accuracy, to assess the classification task. This metric computes the similarity of the pixels between the predicted and ground truth images. For each corresponding pixel pair, if the ratio between the predicted and ground truth values remains under a specific threshold, the classification is deemed accurate. The overall accuracy score is then computed as the percentage of the correctly classified pixels relative to the total number of pixels within the mask. For the show-cased results in Table 4.6 we applied a threshold equal to 1.25. Based on the findings presented in Eq.4.14, it is evident that the models exhibiting the poorest performance solely rely on the edge loss function. Conversely, those



**Figure 4.9:** Depth estimation visualization for L1, Berhu and multi loss respectively

achieving higher accuracy levels are the models that integrate a combination of the Berhu and L1 loss functions.



**Figure 4.10:** Depth estimation ground truth and prediction mask for each classification bin

#### 4.4.5 Comparison of NEFELI’s Pipeline with State-of-the-Art Methods

NEFELI’s tracking module stands out by integrating a deep learning-based appearance model with a Kalman Filter-based motion model, significantly reducing ID switches in the tracking of aerial objects. This is crucial for effective collision avoidance. In contrast to leading methods such as those in [150, 151], which depend solely on Kalman filter-based motion models, NEFELI capitalizes on the combined capabilities of both appearance and motion models.

**Table 4.6:** Classification metrics for the different losses

| Model | Loss functions     | Accuracy Metrics |      |      |                    |
|-------|--------------------|------------------|------|------|--------------------|
|       |                    | Sliding Window   |      |      | Threshold Accuracy |
|       |                    | Mean             | Max  | Min  |                    |
| Unet  | Edge               | 0.14             | 0.14 | 0.14 | 0.01               |
|       | L1                 | 0.61             | 0.16 | 0.71 | 0.86               |
|       | Berhu              | 0.64             | 0.17 | 0.76 | 0.86               |
|       | L1/Berhu           | 0.66             | 0.18 | 0.74 | 0.89               |
|       | Edge/SSIM/L1/Berhu | 0.63             | 0.23 | 0.72 | 0.86               |



The Kalman filter excels in environments where an object’s motion is approximately linear and the measurement noise is Gaussian. However, these conditions are often not met in complex air-to-air detection and tracking scenarios. For example, AirTrack [162] utilizes CenterTrack for air-to-air object detection and tracking, representing objects as center points and relying only on distance offsets between frames for tracking. NEFELI, on the other hand, merges a deep learning appearance model, trained on a specialized re-identification dataset, with a motion model that incorporates data from multiple frames and includes low-confidence detections. This hybrid approach reduces ID switches and improves tracking accuracy and reliability.

## 4.5 Discussion of Real-World Experimental Results

NEFELI enables real-time detection and tracking of non-cooperative aircraft, a vital aspect of autonomous navigation. Consequently, the architecture selection for NEFELI focuses on minimizing latency, eliminating single points of failure, and maximizing system resilience. To determine the optimal architecture, a thorough comparison of cloud versus edge computing was conducted, evaluating their effectiveness in meeting these critical requirements.

Cloud computing transmits data from aircraft sensors to remote servers, introducing round-trip latency that can impair real-time navigation. Conversely, edge computing processes data directly on the aircraft, dramatically reducing transmission delays. Additionally, cloud computing systems are vulnerable to network failures and cloud infrastructure outages, leading to potential operational downtimes with significant risks. Edge computing, however, ensures continued functionality during network disruptions, as processing occurs locally on the aircraft, independent of constant cloud connectivity.

Thus, edge computing is the preferred architecture for NEFELI, offering essential real-time processing capabilities and the robustness needed for

mission-critical operations.

### 4.5.1 Edge Implementation

Deploying NEFELI’s complex machine learning models on low Size, Weight, and Power (Low-SWaP) edge devices is a challenging task. The first step involved selecting the most appropriate edge device for NEFELI. This was achieved by comparing various edge computing options, including Edge GPUs like NVIDIA Jetson, Vision Processing Units (VPUs) such as the Intel Neural Compute Stick (NCS2), and Field-Programmable Gate Arrays (FPGAs). The evaluation criteria included power consumption, performance, scalability, development time, and flexibility [220–222].

The comparison indicated that while VPUs and FPGAs have distinct advantages in power efficiency and hardware customization respectively, NVIDIA Jetson GPUs stood out as the best choice for computer vision tasks at the edge. Their superior performance, power efficiency, scalability, rapid development time, and exceptional flexibility make them the ideal platform for deploying sophisticated computer vision workloads in edge computing environments.

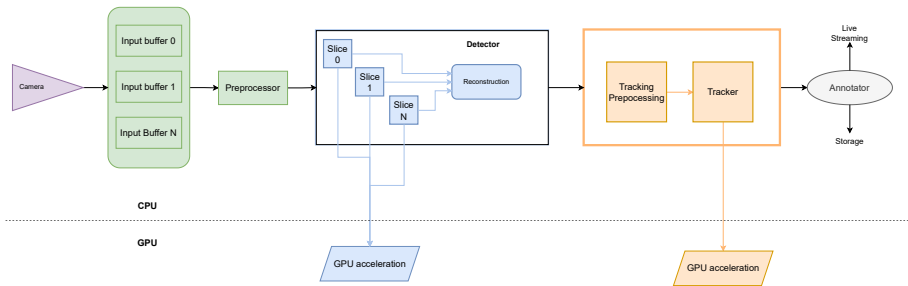


Figure 4.11: NEFELI Edge Implementation

The implementation leverages a streamlined pipeline as shown in Figure 4.11. To reduce latency, given that NEFELI processes data slower than

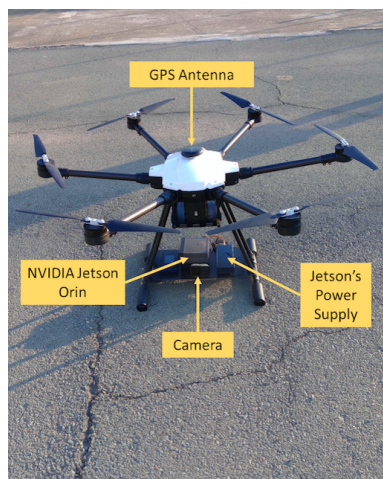
the camera’s capture rate, the input image from the camera is stored in multiple ring buffers. The image undergoes preprocessing steps such as color correction, channel reordering, and slicing into smaller sub-images. These preprocessed images are then fed into a detector optimized by the edge GPU to accelerate inference time. After processing all slices, the image is reassembled to its original size. The tracker module then selects the detected object’s bounding box and uses the Strong-SORT-ReID tracker, also GPU-accelerated, to track it. The final step involves annotating the original image with the bounding box and track ID before transmitting it over the network and saving the output frame.

All CPU-intensive tasks are accelerated using optimized libraries like OpenCV and BLAS, while GPU acceleration is handled by the Open Neural Network Exchange (ONNX) framework. ONNX was chosen for its framework-agnostic design, allowing seamless interoperability and flexibility from cloud training to edge deployment. For ONNX backend execution, TensorRT was selected over CUDA due to its demonstrated superior performance in both literature [223, 224] and experimental results.

### **4.5.2 Evaluation on Real-World Experiments**

To evaluate NEFELI’s performance on an edge GPU and its robustness in real-world scenarios, a series of experiments was conducted. These experiments involved installing the NEFELI system on a small UAV (ownship) and using an identical UAV to simulate the intruder. Both UAVs were hexacopters with a maximum dimension of 1.9 meters, equipped with a Pixhawk 5X flight controller, Pixhawk M9N GPS antenna, and Ardupilot autopilot software [225], as illustrated in Figure 4.12.

NEFELI was deployed on an NVIDIA Jetson Orin edge GPU, utilizing a GoPro Hero 8 camera (Figure 4.12). Importantly, the hardware implementing NEFELI was independent of the UAV’s flight controller, powered separately to ensure no interference with flight operations, thus maintaining flight safety.



**Figure 4.12:** Overview of the components to implement NEFELI

The experiments aimed to test NEFELI's performance under various operational conditions. Different scenarios were tested, including varying angles of view, relative distances, altitudes, lighting conditions, and environmental backgrounds, both above and below the horizon (Figure 4.13).



**Figure 4.13:** Detection results in diverse conditions (below the horizon, close distance, intense lighting, left to right)

**Table 4.7:** Ablation study on NEFELI’s key components using real-flight data. Each component significantly improves performance, demonstrating NEFELI’s robustness on data differing from the training set.

| Method  | HOTA $\uparrow$ | IDs $\downarrow$ | AssA $\uparrow$ | DetA $\uparrow$ | FPPI $\downarrow$ | Speed in fps $\uparrow$ |
|---|-----------------|------------------|-----------------|-----------------|-------------------|-------------------------|
| Baseline  | 23.05           | 2                | 35.61           | 14.98           | <b>0.066</b>      | <b>20.1</b>             |
| Baseline + Sliced Inference (SI)                              | 29.77           | 4                | 37.88           | 23.41           | 0.102             | 7.2                     |
| Baseline + SI + RE-ID (OSnet without EFDM)                    | 34.72           | 1                | 48.29           | 25.02           | 0.099             | 6.7                     |
| <b>NEFELI (Ours)</b>  | <b>37.56</b>    | <b>1</b>         | <b>52.64</b>    | <b>26.80</b>    | 0.097             | 6.7                     |
| (Baseline + Sliced Inference + Re-ID model (OSNet with EFDM)) |                 |                  |                 |                 |                   |                         |

Table 4.7 details NEFELI’s performance for the processing unit and camera used, highlighting improvements from each innovative component introduced. The baseline model used the YOLOv5 detector (without sliced window inference) and the StrongSORT tracker with only motion estimation (without the Re-ID model). Below, the contributions of each component in the NEFELI pipeline are presented through ablation studies.

Ablation study on the effect of sliced inference: The DetA metric in the second row of Table 4.7 demonstrates the efficacy of the proposed sliced inference, compared to no slicing. This postprocessing step effectively enhances the detector’s ability to localize and classify distant, small UAV intruders in real-world experiments.

Ablation study on the effect of the re-identification appearance model: Section 4.4 compares NEFELI’s detection and tracking module using only motion (row 2 in Table 4.7) versus the fused tracker combining motion and appearance models (rows 3 and 4 in Table 4.7). The tracking metrics AssA and IDsw show that the fused approach yields superior results, expected as the appearance model provides additional data-driven information. Using OSNet with Exact Feature Distribution Matching (EFDM) further enhances the model’s ability to reliably track objects in varied real-flight scenarios.

Finally, the maximum distance at which NEFELI can track an intruder was assessed. The theoretical maximum distance, assuming a minimum detectable object size of 12x12 pixels and defining a valid track as detection in four consecutive frames, was 157.9 meters. Experimental results showed successful tracking at distances up to 145.7 meters, aligning closely

**Table 4.8:** Evaluation results (measured in Average Precision (AP)) on the AOT test set and our real-world flight tests using corruptions as augmentations (Finetuned) and without (Base).

| Method    | Dataset      | YOLOv5      | YOLOv8 | YOLOX       | RetinaNet | Faster R-CNN | DiffusionDet | DETR | CenterNet2  |
|-----------|--------------|-------------|--------|-------------|-----------|--------------|--------------|------|-------------|
| Base      | AOT          | 64.6        | 56.4   | <b>69.3</b> | 35.7      | 52.9         | 63.8         | 58.7 | 66.2        |
| Finetuned | AOT          | 65.6        | 56.1   | <b>66.4</b> | 36.1      | 51.2         | 62.4         | 58.1 | 64.3        |
| Base      | real flights | 37.3        | 24.8   | 32.7        | 14.2      | 29.0         | 28.5         | 33.4 | <b>38.9</b> |
| Finetuned | real flights | <b>48.6</b> | 30.4   | 37.9        | 16.7      | 32.9         | 29.3         | 39.1 | 42.3        |

with theoretical expectations. These findings validate NEFELI as a high-performance, vision-based detection and tracking system. Future work will expand these tests with different aircraft types to further evaluate and enhance system capabilities.

### 4.5.3 Enhancing Generalization to Real Flights through Fine-Tuning on Synthetic Corruptions Data

To improve the generalization of object detection models for real-world flight scenarios, we employ a fine-tuning technique using synthetic corruptions as data augmentations. This method introduces diverse and challenging conditions, such as adverse weather, noise, and defocus, into the training process. By exposing the models to these synthetic variations, they learn to adapt to a broader range of environmental factors, enhancing their robustness and performance in diverse and unforeseen conditions encountered during real-world flight tests.

Table 4.8 presents the performance of object detectors evaluated on the clean test set of AOT. The results compare models trained with original datasets (first row) against those fine-tuned with synthetic corruptions (second row). It is evident that the fine-tuned models maintain comparable performance on the same domain (AOT test set), indicating that synthetic corruptions do not negatively impact performance within the same dataset domain.

Additionally, we assessed the fine-tuned and base object detectors on real-world flight tests under varying weather conditions such as cloudy skies,

light rain, and different lighting scenarios. As shown in Table 4.7, models trained with synthetic corruptions (fourth row) demonstrate greater robustness compared to their base counterparts (third row). Notably, the fine-tuned YOLOv5 model shows an 11.3

#### **4.5.4 Correlation between Robustness to Corruptions and Model Generalization**

Robustness to corruptions is intrinsically linked to the generalization capability of object detection models. Generalization denotes a model’s proficiency in performing well on new, unseen data, which includes the ability to manage variations and challenges encountered in real-world environments. Robustness to corruptions measures how effectively a model preserves its performance despite unexpected variations, such as adverse weather, noise, or other distortions.

This correlation is exemplified by comparing the  $AP_{cor}$  values in Table 4.1 with the performance of base models in real-world flight tests (third row) shown in Table 4.8. Object detectors like YOLOv5 and YOLOX, which exhibit strong performance on the AOT-C dataset, also demonstrate robust generalization to challenging real-world flight conditions. An interesting deviation from this pattern is CenterNet, which, despite not excelling on corrupted data, surpasses other detectors in real-world flight tests.

## **4.6 Discussion of Limitations and Failure Cases**

NEFELI aims to advance autonomous aircraft missions by enabling automatic detection and tracking of non-cooperative aircraft. For NEFELI to serve as an effective detect-and-avoid system, several components must be integrated into its pipeline. This section highlights current limitations and identifies key innovations needed to enhance the detection and tracking



models, ensuring a robust and reliable computer vision system for non-cooperative aircraft. Below is a non-exhaustive list of areas for future improvement:

1. **Sensitivity to Extreme Environmental Conditions and Sensor Noise:** NEFELI's performance can degrade under adverse weather conditions such as heavy rain, fog, or snow, which impair visibility and detection accuracy. Tests conducted in such conditions showed decreased performance, indicating a need for improved models and enriched datasets to address these challenges. Sensor noise, especially from cameras, can impact the algorithm's robustness. Real-world sensors may experience various types of noise or degradation, leading to reduced detection performance. The algorithm currently assumes high-quality and reliable sensor operation, but any degradation can adversely affect detection accuracy. Diverse backgrounds, such as complex industrial environments, pose additional challenges. NEFELI was primarily trained on data with specific backgrounds, and its performance may not generalize well to significantly different settings.
2. **Limited Field of View:** The current implementation of NEFELI relies on a limited field of view, potentially missing obstacles or intruders outside this range. This limitation can result in blind spots where potential hazards go undetected. To ensure comprehensive obstacle and intruder detection, a 360-degree field of view is necessary. Integrating multiple sensors to provide a full surround view would enhance NEFELI's capability to detect and track objects from all directions, reducing the risk of collisions.

By discussing these limitations and failure cases, we aim to provide a comprehensive understanding of NEFELI's current capabilities and highlight potential areas for future enhancement.

## **4.7 Summary and Concluding Remarks**

NEFELI represents a pioneering deep-learning approach for automatic aircraft detection and tracking, with each component of the proposed pipeline undergoing rigorous validation. The system’s real-world performance has been thoroughly assessed, and the detection and tracking models have been evaluated on benchmark datasets using standard machine learning metrics. These evaluations position NEFELI as a robust and powerful system for air-to-air aircraft detection and tracking.

Key innovations introduced by NEFELI in both detection and tracking modules are designed for efficient implementation on an edge GPU. The detection module incorporates a sliced inference technique that significantly enhances detection accuracy. This technique allows the processing of high-resolution images (3000 x 4000 pixels) on models trained with lower-resolution images (640 x 640 pixels) without information loss, enabling aircraft detection over longer distances.

The tracking module presents two significant innovations. First, it introduces a large-scale re-identification (Re-ID) dataset for training an appearance model for aircraft tracking. Comparative analyses show that Re-ID versions of the tracking models outperform those not trained on the Re-ID dataset. Second, a novel tracking module combines a Re-ID appearance model for high-confidence detections with a Kalman-filter-based motion model for low-confidence detections, ensuring tracking of even distantly detected aircraft with low confidence.

In addition to these deep learning innovations, NEFELI features an optimized software architecture that enables the implementation of demanding computer vision models on an edge GPU. Real-world experiments validate NEFELI’s capability to detect and track small UAVs at distances of up to 145 meters.

The ultimate aim of NEFELI is to provide essential information to the aircraft’s control system for avoiding mid-air collisions with non-cooperative aircraft. Planned enhancements include the incorporation of a collision

estimation module based on monocular camera distance estimation from tracked aircraft. This module will provide critical information for projecting the movement of non-cooperative aircraft.

Future versions of NEFELI will expand the system’s operational domain to include the intruder’s trajectory projection and automatic maneuvering. Additionally, model uncertainty quantification is crucial for explainability, aligning with aviation standardization bodies’ requirements for AI products in the industry to gain certification.

In conclusion, NEFELI has established a strong foundation, with ongoing efforts focused on enhancing capabilities, increasing system robustness, and addressing challenges posed by adverse environmental conditions.

Another significant contribution of this thesis is the extensive experimental evaluation involving eight diverse object detectors to explore performance degradation under escalating levels of corruptions (domain shifts). Key observations include: 1) One-stage detectors of the YOLO family demonstrate better robustness, 2) Transformer-based and multi-stage detectors like Faster R-CNN are extremely vulnerable to corruptions, and 3) Robustness against corruptions is related to the generalization ability of models. Additionally, fine-tuning on augmented synthetic data results in improvements in the generalization ability of object detectors in real-world flight experiments.



# Chapter 5

## Conclusion

This PhD thesis has addressed pivotal challenges in image recognition, healthcare AI applications, and aircraft detection and tracking, presenting innovative solutions and advanced system architectures that significantly push the boundaries in these domains, particularly in terms of out-of-distribution robustness.

In the field of image recognition, this research introduced the Contrastive Uncertainty Domain Generalisation Network (CUDGNet). This cutting-edge model enhances performance on unseen domains by expanding the source capacity through a fictitious domain generator and utilizing contrastive learning to achieve domain-invariant representations for each class. Extensive experiments on Single Source Domain Generalisation (SSDG) datasets demonstrated that CUDGNet outperforms existing single-DG methods by up to 7.08%. Moreover, CUDGNet offers efficient uncertainty estimation at inference time via a single forward pass through the generator subnetwork, highlighting its practical applicability in real-world scenarios. This contribution advances the robustness of image recognition models when faced with diverse and unfamiliar domains, underscoring the potential of CUDGNet in achieving reliable and generalizable performance.

In the healthcare AI sector, the thesis presented a sophisticated system architecture for the rapid, secure, and scalable deployment of AI applications across heterogeneous computational environments. Central to this architecture is the RACNet model for COVID-19 detection, which provides

healthcare providers with an intuitive, end-to-end interface for uploading DICOM images and receiving timely diagnostic outcomes accompanied by detailed explanations validated by RACNet’s decision-making process. Future work will focus on leveraging user feedback to refine and enhance RACNet’s performance through iterative training and validation. This ongoing evolution aims to bolster diagnostic accuracy, adaptability to emerging clinical challenges, and overall user satisfaction within healthcare settings, ensuring that the system remains at the forefront of medical AI applications.

The thesis also introduced NEFELI, a novel deep-learning approach for automatic aircraft detection and tracking. Each component of the proposed NEFELI pipeline underwent rigorous validation, demonstrating superior performance under real-world conditions. The detection and tracking models were evaluated on benchmark datasets using standard machine learning metrics, positioning the NEFELI pipeline as a powerful system for air-to-air aircraft detection and tracking. The detection module incorporates a sliced inference technique that significantly enhances detection accuracy, allowing the processing of high-resolution images (3000 x 4000 pixels) on models trained with lower-resolution images (640 x 640 pixels) without information loss. This capability enables aircraft detection over much longer distances.

The tracking module presents two significant innovations. Firstly, a large-scale re-identification dataset was created and used to train an appearance model for aircraft tracking, showing that Re-ID versions of the tracking models outperform those not trained on the Re-ID dataset. Secondly, the novel tracking module combines a re-identification appearance model for high-confidence detections and a Kalman-filter-based motion model for low-confidence detections, ensuring that even distantly detected aircraft with low confidence are tracked. Apart from these deep learning innovations, this work presents an optimized software architecture that allows the implementation of demanding computer vision models on an edge GPU. Real-world experiments validate NEFELI’s capability to detect and track small UAVs at distances of 145 meters.

NEFELI aims to provide essential information to the aircraft’s control system for making decisions to avoid mid-air collisions with non-cooperative aircraft. Planned future enhancements include the incorporation of a collision estimation module based on monocular camera distance estimation from tracked aircraft, providing critical information for projecting the movement of non-cooperative aircraft. Future versions of NEFELI will also expand the system’s operational domain to include intruder trajectory projection and automatic maneuvers. Additionally, model uncertainty quantification is crucial for explainability, aligning with aviation standardization bodies’ requirements for AI products in the industry to gain certification.

In summary, this thesis has laid a strong foundation across multiple domains, demonstrating significant advancements in model robustness, system architecture, and real-world applicability. The ongoing efforts focus on enhancing capabilities, increasing system robustness, and addressing challenges posed by adverse conditions, ensuring the continued relevance and impact of the proposed solutions. This work contributes to the broader field of AI by providing robust, scalable, and practical solutions for critical real-world problems, highlighting the potential for future research and development in these areas.

## **5.1 Future Work**

Moving forward, several key areas of research and development are identified to further advance the robustness, efficiency, and applicability of the developed systems.

In the domain of image recognition, future efforts will focus on enhancing model explainability, particularly for transformer-based networks. Exploring methods to interpret and visualize the decision-making process of these models will be crucial to build trust and transparency in their outputs across various domains and applications.

In healthcare AI, leveraging user feedback to iteratively refine and enhance RACNet’s diagnostic capabilities will remain a cornerstone. This iterative improvement process will involve gathering user insights from healthcare providers to enhance diagnostic accuracy and usability. Concurrently, developing robust uncertainty quantification techniques and improving model explainability will be critical to validate and justify RACNet’s decisions in clinical settings.

For NEFELI, the future research will aim to develop a more generalizable object detection framework capable of adapting to diverse operational environments and conditions. This includes augmenting training datasets with a wider range of scenarios and environmental conditions to improve model robustness and generalization. Additionally, developing an efficient depth-distance estimation model integrated with NEFELI’s pipeline will enhance its ability to accurately assess the proximity of detected objects, crucial for collision avoidance strategies.

Exploring advanced collision avoidance algorithms that integrate seamlessly with NEFELI’s detection and tracking capabilities will be essential. These algorithms will not only predict potential collisions but also autonomously recommend or execute evasive maneuvers to ensure the safety of manned and unmanned aircraft.

Furthermore, ensuring scalability and deployment readiness of these systems across different platforms and operational scenarios will be a priority. This involves optimizing software architectures for edge devices and cloud-based deployments while adhering to industry standards and regulatory requirements in aviation and healthcare.

By addressing these future directions, the research aims to advance the state-of-the-art in AI applications, particularly in image recognition, healthcare AI, and autonomous systems for aircraft detection and collision avoidance, fostering safer and more reliable technological solutions in mission-critical domains.



# Γλωσσάρι

|  |   |
|--|---|
| Adversarial Data Augmentation                      | Η διαδικασία περιλαμβάνει την παραγωγή νέων δεδομένων από υπάρχοντα δεδομένα μέσω της προσθήκης εχθρικών παραδειγμάτων. Αυτά τα εχθρικά παραδείγματα είναι τεχνητά δεδομένα που έχουν δημιουργηθεί με σκοπό να προκαλέσουν σφάλματα στους αλγόριθμους, βοηθώντας έτσι στην ανθεκτικότητα και την ικανότητα γενίκευσης των μοντέλων. |
| Artificial intelligence (AI)                       | Τεχνητή Νοημοσύνη.  |
| Anchor Set   | Σημεία αγκύρωσης για την ομαδοποίηση δεδομένων. Αυτά τα σημεία βοηθούν στη σταθεροποίηση και βελτίωση της διαδικασίας ομαδοποίησης..  |
| Airborne Collision Avoidance System (ACAS)         | Σύστημα Αποφυγής Εναέριων Συγκρούσεων.  |
| Automatic Dependent Surveillance-Broadcast (ADS-B) | Αυτόματη Εξαρτώμενη Επιτήρηση-Μετάδοση.   |
| Air-to-Air Object Detection                        | Ανίχνευση αντικειμένων από αέρος σε αέρα.   |

|   |  |
|---|--|
| Aerial Object Tracking  | Ανίχνευση και παρακολούθηση αντικειμένων από αέρος.  |
| Bayesian meta-learning  | Μπευζιανή Μετα-μάθηση.   |
| Contrastive learning  | Αντιθετική μάθηση είναι μια τεχνική στη μηχανική μάθηση που χρησιμοποιείται για την εκμάθηση χρήσιμων αναπαραστάσεων δεδομένων συγκρίνοντας διαφορετικά ζεύγη παραδειγμάτων.         |
| Convolutional Neural Network (CNN)                            | Συνελικτικό Νευρωνικό Δίκτυο.  |
| Domain Adversarial Learning                                   | Διαφορετική Μάθηση μέσω Αντιπαλότητας είναι μια τεχνική μηχανικής μάθησης που χρησιμοποιείται για να βελτιώσει την ικανότητα γενίκευσης ενός μοντέλου σε διάφορους τομείς (δομains). |
| Domain Generalization (DG)                                    | Γενίκευση Τομέα.   |
| Digital Imaging and Communications in Medicine (DICOM) format | Ένα πρότυπο που χρησιμοποιείται ευρέως στον τομέα της ιατρικής απεικόνισης για την αποθήκευση, την αναμετάδοση και την επεξεργασία ιατρικών εικόνων και σχετικών δεδομένων..         |
| High-Performance Computing (HPC)                              | Αναφέρεται σε τεχνολογίες και τεχνικές που χρησιμοποιούνται για την επίτευξη εξαιρετικά υψηλών επιδόσεων υπολογισμού,.   |
| Medical image analysis (MedIA)                                | Ανάλυση ιατρικών εικόνων.  |

|  |  |
|--|--|
| Microservices architecture                 | Αρχιτεκτονική μικροϋπηρεσιών είναι ένα στυλ αρχιτεκτονικής λογισμικού που διαχωρίζει μια εφαρμογή σε μικρές, αυτόνομες υπηρεσίες που επικοινωνούν μεταξύ τους μέσω καλά καθορισμένων διεπαφών. |
| Mid-air collision (MAC)                    | Σύγκρουση στον αέρα.   |
| Monocular Distance Estimation              | Μονόφθαλμη εκτίμηση απόστασης αναφέρεται στη διαδικασία υπολογισμού της απόστασης ενός αντικειμένου από τον παρατηρητή χρησιμοποιώντας μια μόνο κάμερα ή έναν μόνο φακό.                       |
| Out-of-distribution (OOD)                  | Εκτός κατανομής.   |
| Re-identification Model                    | Μοντέλο επαναταυτοποίησης.   |
| Recurrent Neural Network (RNN)             | Επαναλαμβανόμενο Νευρωνικό Δίκτυο.   |
| Single-Source Domain Generalization (SSDG) | Γενίκευση Τομέα από μία μόνο πηγή.   |
| Style transfer                             | Είναι μια τεχνική στην υπολογιστική όραση και στην τεχνητή νοημοσύνη που επιτρέπει τη μεταφορά του στυλ ενός εικόνας σε μια άλλη, ενώ ταυτόχρονα διατηρεί το περιεχόμενο της δεύτερης εικόνας. |

|                                   |  |
|-----------------------------------|--|
| Synthetic Common Corruptions      | Δημιουργία ρεαλιστικών και ποικιλόμορφων συνθηκών με παραμορφώσεις για εκπαίδευση και αξιολόγηση, βελτιώνοντας τη γενική απόδοση των συστημάτων μηχανικής μάθησης και αναγνωστικών μοντέλων. |
| Transformation Component          | Συστατικό Μετασχηματισμού.   |
| UI/UX                             | Αναφέρεται σε δύο αλληλένδετες πτυχές του σχεδιασμού ψηφιακών προϊόντων και υπηρεσιών, κυρίως εφαρμογών και ιστοσελίδων.   |
| Uncertainty Estimation            | Εκτίμηση Αβεβαιότητας.   |
| Unmanned aerial vehicle (UAV)     | Μη-επανδρωμένο αεροσκάφος.   |
| Urban Advanced Air Mobility (AAM) | Αναφέρεται στην εφαρμογή προηγμένων τεχνολογιών αεροπορίας για τη βελτίωση των μεταφορών και της κινητικότητας μέσα σε αστικά περιβάλλοντα..   |

# Publications

## Journals

1. Kollias, D., Arsenos, A., Kollias, S. (2023). A deep neural architecture for harmonizing 3-d input data analysis and decision making in medical imaging. *Neurocomputing*, 542, 126244.
2. Arsenos, A., Karampinis, V., Petrongonas, E., Skliros, C., Kollias, D., Kollias, S., & Voulodimos, A. (2024). Common Corruptions for Evaluating and Enhancing Robustness in Air-to-Air Visual Object Detection. *IEEE Robotics and Automation Letters*.
3. Arsenos, A., Petrongonas, E., Filippopoulos, O., Skliros, C., Kollias, D., & Kollias, S. Nefeli: A deep-learning detection and tracking pipeline for enhancing autonomy in advanced air mobility. Available at SSRN 4674579.

## Conferences

1. Arsenos, A., Kollias, D., Petrongonas, E., Skliros, C., Kollias, S. (2024, April). Uncertainty-guided contrastive learning for single source domain generalisation. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6935-6939). IEEE.

2. Gerogiannis, D., Arsenos, A., Kollias, D., Nikitopoulos, D., Kollias, S. (2024). Covid-19 computer-aided diagnosis through ai-assisted ct imaging analysis: Deploying a medical ai system. arXiv preprint arXiv:2403.06242. ISBI 2024
3. Karampinis, V., Arsenos, A., Filippopoulos, O., Petrongonas, E., Skliros, C., Kollias, D., ... Voulodimos, A. (2024). Ensuring UAV Safety: A Vision-only and Real-time Framework for Collision Avoidance Through Object Detection, Tracking, and Distance Estimation. arXiv preprint arXiv:2405.06749. ICUAS 2024

## Workshops

1. Kollias, D., Arsenos, A., Soukissian, L., Kollias, S. (2021). Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 537-544).
2. Arsenos, A., Kollias, D., Kollias, S. (2022, June). A large imaging database and novel deep neural architecture for covid-19 diagnosis. In 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP) (pp. 1-5). IEEE.
3. Kollias, D., Arsenos, A., Kollias, S. (2022, October). Ai-mia: Covid-19 detection and severity analysis through medical imaging. In European Conference on Computer Vision (pp. 677-690). Cham: Springer Nature Switzerland.
4. Kollias, D., Arsenos, A., Kollias, S. (2023, June). Ai-enabled analysis of 3-d ct scans for diagnosis of covid-19 its severity. In 2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW) (pp. 1-5). IEEE.
5. Arsenos, A., Davidhi, A., Kollias, D., Prassopoulos, P., Kollias, S.

- (2023, June). Data-driven covid-19 detection through medical imaging. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)* (pp. 1-5). IEEE.
6. Kollias, D., Arsenos, A., Kollias, S. (2024). Domain Adaptation Explainability Fairness in AI for Medical Image Analysis: Diagnosis of COVID-19 based on 3-D Chest CT-scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4907-4914).
  7. Spanos, N., Arsenos, A., Theofilou, P. A., Tzouveli, P., Voulodimos, A., Kollias, S. (2024). Complex Style Image Transformations for Domain Generalization in Medical Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5036-5045).





## Bibliography

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [2] G. Blanchard, G. Lee, and C. Scott, “Generalizing from several related classification tasks to a new unlabeled sample,” *Advances in neural information processing systems*, vol. 24, 2011.
- [3] X. Fan, Q. Wang, J. Ke, F. Yang, B. Gong, and M. Zhou, “Adversarially adaptive normalization for single domain generalization,” in *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 8208–8217, 2021.
- [4] X. Zhang, P. Cui, R. Xu, L. Zhou, Y. He, and Z. Shen, “Deep stable learning for out-of-distribution generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5372–5382, 2021.
- [5] P. Pandey, M. Raman, S. Varambally, and P. AP, “Domain generalization via inference-time label-preserving target projections,” *arXiv preprint arXiv:2103.01134*, 2021.
- [6] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, “Open domain generalization with domain-augmented meta-learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9624–9633, 2021.
- [7] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, “Mixstyle neural networks for domain generalization and adaptation,” *International Journal of Computer Vision*, pp. 1–15, 2023.
- [8] K. Zhou, C. C. Loy, and Z. Liu, “Semi-supervised domain generalization with stochastic stylematch,” *International Journal of Computer Vision*, pp. 1–11, 2023.
- [9] J. Cha, S. Chun, K. Lee, H.-C. Cho, S. Park, Y. Lee, and S. Park,

- “Swad: Domain generalization by seeking flat minima,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 22405–22418, 2021.
- [10] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018.
- [11] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, “Deep domain generalization via conditional invariant adversarial networks,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639, 2018.
- [12] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales, “Learning to generalize: Meta-learning for domain generalization,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.
- [13] Y. Balaji, S. Sankaranarayanan, and R. Chellappa, “Metareg: Towards domain generalization using meta-regularization,” *Advances in neural information processing systems*, vol. 31, 2018.
- [14] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, “Learning to generate novel domains for domain generalization,” in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pp. 561–578, Springer, 2020.
- [15] K. Zhou, Y. Yang, T. Hospedales, and T. Xiang, “Deep domain-adversarial image generation for domain generalisation,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13025–13032, 2020.
- [16] D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales, “Deeper, broader and artier domain generalization,” in *Proceedings of the IEEE international conference on computer vision*, pp. 5542–5550, 2017.
- [17] Y. Li, Y. Yang, W. Zhou, and T. Hospedales, “Feature-critic networks for heterogeneous domain generalization,” in *International Conference on Machine Learning*, pp. 3915–3924, PMLR, 2019.
- [18] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, “Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data,” in *Proceedings of the IEEE/CVF International Conference*

- 
- on *Computer Vision*, pp. 2100–2110, 2019.
- [19] R. Volpi and V. Murino, “Addressing model vulnerability to distributional shifts over image transformation sets,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7980–7989, 2019.
- [20] K. Zhou, Y. Yang, Y. Qiao, and T. Xiang, “Domain generalization with mixstyle,” *arXiv preprint arXiv:2104.02008*, 2021.
- [21] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, “Generalizing across domains via cross-gradient training,” *arXiv preprint arXiv:1804.10745*, 2018.
- [22] Q. Liu, Q. Dou, L. Yu, and P. A. Heng, “Ms-net: multi-site network for improving prostate segmentation with heterogeneous mri data,” *IEEE transactions on medical imaging*, vol. 39, no. 9, pp. 2713–2724, 2020.
- [23] Q. Liu, Q. Dou, and P.-A. Heng, “Shape-aware meta-learning for generalizing prostate mri segmentation to unseen domains,” in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part II 23*, pp. 475–485, Springer, 2020.
- [24] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.
- [25] N. Honarvar Nazari and A. Kovashka, “Domain generalization using shape representation,” in *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pp. 666–670, Springer, 2020.
- [26] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, IEEE, 2017.
- [27] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 3803–3810, IEEE, 2018.
- [28] R. Khirodkar, D. Yoo, and K. Kitani, “Domain randomization for

- scene-specific car detection and pose estimation,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1932–1940, IEEE, 2019.
- [29] X. Yue, Y. Zhang, S. Zhao, A. Sangiovanni-Vincentelli, K. Keutzer, and B. Gong, “Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2100–2110, 2019.
- [30] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, “Structured domain randomization: Bridging the reality gap by context-aware synthetic data,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7249–7255, IEEE, 2019.
- [31] G. Blanchard, A. A. Deshmukh, U. Dogan, G. Lee, and C. Scott, “Domain generalization by marginal transfer learning,” *Journal of machine learning research*, vol. 22, no. 2, pp. 1–55, 2021.
- [32] S. Shankar, V. Piratla, S. Chakrabarti, S. Chaudhuri, P. Jyothi, and S. Sarawagi, “Generalizing across domains via cross-gradient training,” *arXiv preprint arXiv:1804.10745*, 2018.
- [33] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, “Generalizing to unseen domains via adversarial data augmentation,” *Advances in neural information processing systems*, vol. 31, 2018.
- [34] J. Huang, D. Guan, A. Xiao, and S. Lu, “Fsdr: Frequency space domain randomization for domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6891–6902, 2021.
- [35] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.

- 
- [38] L. Li, K. Gao, J. Cao, Z. Huang, Y. Weng, X. Mi, Z. Yu, X. Li, and B. Xia, “Progressive domain expansion network for single domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 224–233, 2021.
- [39] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A kernel two-sample test,” *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.
- [40] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schölkopf, “Wasserstein auto-encoders,” *arXiv preprint arXiv:1711.01558*, 2017.
- [41] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker, “Domain generalization via model-agnostic learning of semantic features,” *Advances in neural information processing systems*, vol. 32, 2019.
- [42] H. Li, S. J. Pan, S. Wang, and A. C. Kot, “Domain generalization with adversarial feature learning,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5400–5409, 2018.
- [43] Y. Shu, Z. Cao, C. Wang, J. Wang, and M. Long, “Open domain generalization with domain-augmented meta-learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9624–9633, 2021.
- [44] F. Qiao and X. Peng, “Uncertainty-guided model generalization to unseen domains,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6790–6800, 2021.
- [45] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [46] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, pp. 273–297, 1995.
- [47] C. Gan, T. Yang, and B. Gong, “Learning attributes equals multi-source domain generalization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 87–97, 2016.
- [48] T. Grubinger, A. Birlutiu, H. Schöner, T. Natschläger, and T. Heskes, “Domain generalization based on transfer component analysis,” in *Advances in Computational Intelligence: 13th International Work-Conference on Artificial Neural Networks, IWANN 2015, Palma*

- de Mallorca, Spain, June 10-12, 2015. Proceedings, Part I 13*, pp. 325–334, Springer, 2015.
- [49] Y. Li, M. Gong, X. Tian, T. Liu, and D. Tao, “Domain generalization via conditional invariant representations,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, 2018.
- [50] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, “Scatter component analysis: A unified framework for domain adaptation and domain generalization,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 7, pp. 1414–1430, 2016.
- [51] S. Erfani, M. Baktashmotlagh, M. Moshtaghi, X. Nguyen, C. Leckie, J. Bailey, and R. Kotagiri, “Robust domain generalisation by enforcing distribution invariance,” in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16)*, pp. 1455–1461, AAAI Press, 2016.
- [52] S. Hu, K. Zhang, Z. Chen, and L. Chan, “Domain generalization via multidomain discriminant analysis,” in *Uncertainty in Artificial Intelligence*, pp. 292–302, PMLR, 2020.
- [53] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *International conference on machine learning*, pp. 1180–1189, PMLR, 2015.
- [54] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, “Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2097–2106, 2017.
- [55] R. Gong, W. Li, Y. Chen, and L. V. Gool, “Dlow: Domain flow for adaptation and generalization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2477–2486, 2019.
- [56] Y. Li, X. Tian, M. Gong, Y. Liu, T. Liu, K. Zhang, and D. Tao, “Deep domain generalization via conditional invariant adversarial networks,” in *Proceedings of the European conference on computer vision (ECCV)*, pp. 624–639, 2018.
- [57] Y. Jia, J. Zhang, S. Shan, and X. Chen, “Single-side domain generalization for face anti-spoofing,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recog-*

- 
- niton, pp. 8484–8493, 2020.
- [58] X. Pan, P. Luo, J. Shi, and X. Tang, “Two at once: Enhancing learning and generalization capacities via ibn-net,” in *Proceedings of the european conference on computer vision (ECCV)*, pp. 464–479, 2018.
  - [59] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” *arXiv preprint arXiv:1412.3474*, 2014.
  - [60] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.
  - [61] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.
  - [62] F. Zhou, Z. Jiang, C. Shui, B. Wang, and B. Chaib-draa, “Domain generalization via optimal transport with metric similarity learning,” *Neurocomputing*, vol. 456, pp. 469–480, 2021.
  - [63] J. Jia, Q. Ruan, and T. M. Hospedales, “Frustratingly easy person re-identification: Generalizing person re-id in practice,” *arXiv preprint arXiv:1905.03422*, 2019.
  - [64] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6924–6932, 2017.
  - [65] X. Jin, C. Lan, W. Zeng, Z. Chen, and L. Zhang, “Style normalization and restitution for generalizable person re-identification,” in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3143–3152, 2020.
  - [66] X. Jin, C. Lan, W. Zeng, and Z. Chen, “Style normalization and restitution for domain generalization and adaptation,” *IEEE Transactions on Multimedia*, vol. 24, pp. 3636–3651, 2021.
  - [67] X. Jin, C. Lan, W. Zeng, and Z. Chen, “Feature alignment and restoration for domain generalization and adaptation,” *arXiv preprint arXiv:2006.12009*, 2020.
  - [68] L. Qi, L. Wang, Y. Shi, and X. Geng, “Unsupervised domain gener-
-

- alization for person re-identification: A domain-specific adaptive framework,” *arXiv preprint arXiv:2111.15077*, 2021.
- [69] E. Luz, P. Silva, R. Silva, L. Silva, J. Guimarães, G. Miozzo, G. Moreira, and D. Menotti, “Towards an effective and efficient deep learning model for covid-19 patterns detection in x-ray images,” *Research on Biomedical Engineering*, pp. 1–14, 2021.
- [70] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” *arXiv preprint arXiv:1907.02893*, 2019.
- [71] K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar, “Invariant risk minimization games,” in *International Conference on Machine Learning*, pp. 145–155, PMLR, 2020.
- [72] E. Rosenfeld, P. Ravikumar, and A. Risteski, “The risks of invariant risk minimization,” *arXiv preprint arXiv:2010.05761*, 2020.
- [73] D. Krueger, E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville, “Out-of-distribution generalization via risk extrapolation (rex),” in *International Conference on Machine Learning*, pp. 5815–5826, PMLR, 2021.
- [74] J. Mitrovic, B. McWilliams, J. Walker, L. Buesing, and C. Blundell, “Representation learning via invariant causal mechanisms,” *arXiv preprint arXiv:2010.07922*, 2020.
- [75] L. Niu, W. Li, and D. Xu, “Multi-view domain generalization for visual recognition,” in *Proceedings of the IEEE international conference on computer vision*, pp. 4193–4201, 2015.
- [76] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba, “Undoing the damage of dataset bias,” in *Computer Vision—ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part I 12*, pp. 158–171, Springer, 2012.
- [77] Z. Ding and Y. Fu, “Deep domain generalization with structured low-rank constraint,” *IEEE Transactions on Image Processing*, vol. 27, no. 1, pp. 304–313, 2017.
- [78] Y. Wang, H. Li, H. Cheng, B. Wen, L.-P. Chau, and A. C. Kot, “Variational disentanglement for domain generalization,” *arXiv preprint arXiv:2109.05826*, 2021.
- [79] M. Ilse, J. M. Tomczak, C. Louizos, and M. Welling, “Diva: Domain invariant variational autoencoders,” in *Medical Imaging with Deep*



- 
- Learning*, pp. 322–348, PMLR, 2020.
- [80] X. Peng, Z. Huang, X. Sun, and K. Saenko, “Domain agnostic learning with disentangled representations,” in *International Conference on Machine Learning*, pp. 5102–5112, PMLR, 2019.
  - [81] F. Qiao, L. Zhao, and X. Peng, “Learning to learn single domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020.
  - [82] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, “Towards principled disentanglement for domain generalization,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8024–8034, 2022.
  - [83] D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, J. M. Mooij, and B. Schölkopf, “On causal and anticausal learning,” in *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1255–1262, 2012.
  - [84] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, “Domain adaptation under target and conditional shift,” in *International conference on machine learning*, pp. 819–827, Pmlr, 2013.
  - [85] K. Zhang, M. Gong, and B. Schölkopf, “Multi-source domain adaptation: A causal view,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.
  - [86] M. Gong, K. Zhang, B. Huang, C. Glymour, D. Tao, and K. Batmanghelich, “Causal generative domain adaptation networks,” *arXiv preprint arXiv:1804.04333*, 2018.
  - [87] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” *Advances in neural information processing systems*, vol. 30, 2017.
  - [88] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *International conference on machine learning*, pp. 1126–1135, PMLR, 2017.
  - [89] K. Muandet, D. Balduzzi, and B. Schölkopf, “Domain generalization via invariant feature representation,” in *International conference on machine learning*, pp. 10–18, PMLR, 2013.
  - [90] A. Dubey, V. Ramanathan, A. Pentland, and D. Mahajan, “Adaptive methods for real-world domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

- Recognition*, pp. 14340–14349, 2021.
- [91] K. Chen, D. Zhuang, and J. M. Chang, “Discriminative adversarial domain generalization with meta-learning based cross-domain validation,” *Neurocomputing*, vol. 467, pp. 418–426, 2022.
  - [92] H. Sharifi-Noghabi, H. Asghari, N. Mehrasa, and M. Ester, “Domain generalization via semi-supervised meta learning,” *arXiv preprint arXiv:2009.12658*, 2020.
  - [93] A. Rame, C. Dancette, and M. Cord, “Fishr: Invariant gradient variances for out-of-distribution generalization,” in *International Conference on Machine Learning*, pp. 18347–18377, PMLR, 2022.
  - [94] C. X. Tian, H. Li, X. Xie, Y. Liu, and S. Wang, “Neuron coverage-guided domain generalization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1302–1311, 2022.
  - [95] G. Wu and S. Gong, “Collaborative optimization and aggregation for decentralized domain generalization and adaptation,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6484–6493, 2021.
  - [96] D. Adila and D. Kang, “Understanding out-of-distribution: A perspective of data dynamics,” in *I (Still) Can’t Believe It’s Not Better! Workshop at NeurIPS 2021*, pp. 1–8, PMLR, 2022.
  - [97] L. Li, K. Gao, J. Cao, Z. Huang, Y. Weng, X. Mi, Z. Yu, X. Li, and B. Xia, “Progressive domain expansion network for single domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 224–233, 2021.
  - [98] Z. Wang, Y. Luo, R. Qiu, Z. Huang, and M. Baktashmotlagh, “Learning to diversify for single domain generalization,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 834–843, 2021.
  - [99] F. M. Carlucci, A. D’Innocente, S. Bucci, B. Caputo, and T. Tommasi, “Domain generalization by solving jigsaw puzzles,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2229–2238, 2019.
  - [100] C. Wan, X. Shen, Y. Zhang, Z. Yin, X. Tian, F. Gao, J. Huang, and X.-S. Hua, “Meta convolutional neural networks for single domain generalization,” in *Proceedings of the IEEE/CVF Conference on*

- 
- Computer Vision and Pattern Recognition*, pp. 4682–4691, 2022.
- [101] F. Qiao, L. Zhao, and X. Peng, “Learning to learn single domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12556–12565, 2020.
- [102] Z. Xu, D. Liu, J. Yang, C. Raffel, and M. Niethammer, “Robust and generalizable visual representation learning via random convolutions,” *ICLR*, 2021.
- [103] X. Peng, F. Qiao, and L. Zhao, “Out-of-domain generalization from a single source: An uncertainty quantification approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [104] K. Hirokatsu, M. Asato, Y. Eisuke, Y. Ryosuke, I. Nakamasa, A. Nakamura, and S. Yutaka, “Pre-training without natural images,” *International Journal of Computer Vision*, vol. 130, no. 4, pp. 990–1007, 2022.
- [105] D. Coltuc, P. Bolon, and J.-M. Chassery, “Exact histogram specification,” *IEEE Transactions on Image processing*, vol. 15, no. 5, pp. 1143–1152, 2006.
- [106] J. P. Rolland, V. Vo, B. Bloss, and C. K. Abbey, “Fast algorithms for histogram matching: Application to texture synthesis,” *Journal of Electronic Imaging*, vol. 9, no. 1, pp. 39–45, 2000.
- [107] X. Chen and K. He, “Exploring simple siamese representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15750–15758, 2021.
- [108] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.
- [109] R. Volpi, H. Namkoong, O. Sener, J. C. Duchi, V. Murino, and S. Savarese, “Generalizing to unseen domains via adversarial data augmentation,” *Advances in neural information processing systems*, vol. 31, 2018.
- [110] T. Gokhale, R. Anirudh, J. J. Thiagarajan, B. Kailkhura, C. Baral, and Y. Yang, “Improving diversity with adversarially learned transformations for domain generalization,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 434–443, 2023.
- [111] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight

- uncertainty in neural network,” in *International conference on machine learning*, pp. 1613–1622, PMLR, 2015.
- [112] Y. Nan, J. Del Ser, S. Walsh, C. Schönlieb, M. Roberts, I. Selby, K. Howard, J. Owen, J. Neville, J. Guiot, *et al.*, “Data harmonisation for information fusion in digital healthcare: A state-of-the-art systematic review, meta-analysis and future research directions,” *Information Fusion*, vol. 82, pp. 99–122, 2022.
- [113] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, and F. Mahmood, “Scaling vision transformers to gigapixel images via hierarchical self-supervised learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16144–16155, 2022.
- [114] L. Feng, S. Coppo, D. Piccini, J. Yerly, R. P. Lim, P. G. Masci, M. Stuber, D. K. Sodickson, and R. Otazo, “5d whole-heart sparse mri,” *Magnetic resonance in medicine*, vol. 79, no. 2, pp. 826–838, 2018.
- [115] D. Kollias, A. Arsenos, L. Soukissian, and S. Kollias, “Mia-cov19d: Covid-19 detection through 3-d chest ct image analysis,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 537–544, 2021.
- [116] D. Kollias, N. Bouas, Y. Vlaxos, V. Brillakis, M. Seferis, I. Kollia, L. Sukissian, J. Wingate, and S. Kollias, “Deep transparent prediction through latent representation analysis,” *arXiv preprint arXiv:2009.07044*, 2020.
- [117] D. Kollias, Y. Vlaxos, M. Seferis, I. Kollia, L. Sukissian, J. Wingate, and S. D. Kollias, “Transparent adaptation in deep medical image diagnosis.,” in *TAILOR*, pp. 251–267, 2020.
- [118] D. Kollias, A. Tagaris, A. Stafylopatis, S. Kollias, and G. Tagaris, “Deep neural architectures for prediction in healthcare,” *Complex & Intelligent Systems*, vol. 4, no. 2, pp. 119–131, 2018.
- [119] D. Kollias, A. Arsenos, and S. Kollias, “Ai-mia: Covid-19 detection and severity analysis through medical imaging,” in *Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pp. 677–690, Springer, 2023.
- [120] A. Arsenos, D. Kollias, and S. Kollias, “A large imaging database and novel deep neural architecture for covid-19 diagnosis,” in *2022 IEEE 14th Image, Video, and Multidimensional Signal Processing*

- 
- Workshop (IVMSP)*, p. 1–5, IEEE, 2022.
- [121] K. Zhang, X. Liu, J. Shen, Z. Li, Y. Sang, X. Wu, Y. Zha, W. Liang, C. Wang, K. Wang, L. Ye, M. Gao, Z. Zhou, L. Li, J. Wang, Z. Yang, H. Cai, J. Xu, L. Yang, W. Cai, W. Xu, S. Wu, W. Zhang, S. Jiang, L. Zheng, X. Zhang, L. Wang, L. Lu, J. Li, H. Yin, W. Wang, O. Li, C. Zhang, L. Liang, T. Wu, R. Deng, K. Wei, Y. Zhou, T. Chen, J. Yiu-Nam Lau, M. Fok, J. He, T. Lin, W. Li, and G. Wang, “Clinically applicable AI system for accurate diagnosis, quantitative measurements, and prognosis of COVID-19 pneumonia using computed tomography,” *Cell*, vol. 182, p. 1360, Sept. 2020.
- [122] X. He, S. Wang, X. Chu, S. Shi, J. Tang, X. Liu, C. Yan, J. Zhang, and G. Ding, “Automated model design and benchmarking of deep learning models for covid-19 detection with chest ct scans,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, pp. 4821–4829, 2021.
- [123] S. P. Morozov, A. E. Andreychenko, I. A. Blokhin, P. B. Gelezhe, A. P. Gonchar, A. E. Nikolaev, N. A. Pavlov, V. Y. Chernina, and V. A. Gombolevskiy, “Mosmeddata: data set of 1110 chest ct scans performed during the covid-19 epidemic,” *Digital Diagnostics*, vol. 1, no. 1, pp. 49–59, 2020.
- [124] J. Zhao, Y. Zhang, X. He, and P. Xie, “Covid-ct-dataset: a ct scan dataset about covid-19,” *arXiv preprint arXiv:2003.13865*, 2020.
- [125] Z. Han, B. Wei, Y. Hong, T. Li, J. Cong, X. Zhu, H. Wei, and W. Zhang, “Accurate screening of covid-19 using attention-based deep 3d multiple instance learning,” *IEEE transactions on medical imaging*, vol. 39, no. 8, pp. 2584–2594, 2020.
- [126] S. Chen, K. Ma, and Y. Zheng, “Med3d: Transfer learning for 3d medical image analysis,” *arXiv preprint arXiv:1904.00625*, 2019.
- [127] C.-C. Hsu, G.-L. Chen, and M.-H. Wu, “Visual transformer with statistical test for covid-19 classification,” 2021.
- [128] R. Miron, C. Moisii, S. Dinu, and M. Breaban, “Covid detection in chest cts: Improving the baseline on cov19-ct-db,” 2021.
- [129] J. Hou, J. Xu, R. Feng, Y. Zhang, F. Shan, and W. Shi, “Cmc-cov19d: Contrastive mixup classification for covid-19 diagnosis,” in *Proceedings of the IEEE/CVF International Conference on Computer*

- Vision (ICCV) Workshops*, pp. 454–461, October 2021.
- [130] R. Turnbull, “Cov3d: Detection of the presence and severity of COVID-19 from CT scans using 3D ResNets [Preliminary Preprint],” 7 2022.
- [131] J. Hou, J. Xu, R. Feng, and Y. Zhang, “Fdvts’s solution for 2nd cov19d competition on covid-19 detection and severity analysis,” 2022.
- [132] C.-C. Hsu, C.-H. Tsai, G.-L. Chen, S.-D. Ma, and S.-C. Tai, “Spatiotemporal feature learning based on two-step lstm and transformer for ct scans,” 2022.
- [133] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” 2017.
- [134] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool, “Temporal 3d convnets: New architecture and transfer learning for video classification,” 2017.
- [135] X. Wu, C. Chen, M. Zhong, J. Wang, and J. Shi, “COVID-AL: The diagnosis of COVID-19 with deep active learning,” *Med Image Anal*, vol. 68, p. 101913, Nov. 2020.
- [136] X. Li, W. Tan, P. Liu, Q. Zhou, and J. Yang, “Classification of covid-19 chest ct images based on ensemble deep learning,” *Journal of Healthcare Engineering*, vol. 2021, p. 5528441, Apr 2021.
- [137] X. He, S. Wang, S. Shi, X. Chu, J. Tang, X. Liu, C. Yan, J. Zhang, and G. Ding, “Benchmarking deep learning models and automated model design for covid-19 detection with chest ct scans,” *medRxiv*, 2021.
- [138] X. He, G. Ying, J. Zhang, and X. Chu, “Evolutionary multi-objective architecture search framework: Application to covid-19 3d ct classification,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022* (L. Wang, Q. Dou, P. T. Fletcher, S. Speidel, and S. Li, eds.), (Cham), pp. 560–570, Springer Nature Switzerland, 2022.
- [139] X. Wang, X. Deng, Q. Fu, Q. Zhou, J. Feng, H. Ma, W. Liu, and C. Zheng, “A weakly-supervised framework for covid-19 classification and lesion localization from chest ct,” *IEEE transactions on medical imaging*, vol. 39, no. 8, pp. 2615–2625, 2020.
- [140] W. Zouch, D. Sagga, A. Ectiou, R. Khemakhem, M. Ghorbel,

- C. Mhiri, and A. B. Hamida, "Detection of covid-19 from ct and chest x-ray images using deep learning models," *Annals of Biomedical Engineering*, vol. 50, pp. 825–835, Jul 2022.
- [141] D. Kollias, Y. Vlaxos, M. Seferis, I. Kollia, L. Sukissian, J. Wingate, and S. D. Kollias, "Transparent adaptation in deep medical image diagnosis.," in *TAILOR*, p. 251–267, 2020.
- [142] A. Arsenos, A. Davidhi, D. Kollias, P. Prassopoulos, and S. Kollias, "Data-driven covid-19 detection through medical imaging," in *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5, IEEE, 2023.
- [143] J. Soldani, D. A. Tamburri, and W.-J. Van Den Heuvel, "The pains and gains of microservices: A systematic grey literature review," *Journal of Systems and Software*, vol. 146, pp. 215–232, 2018.
- [144] Internet Engineering Task Force, "The OAuth 2.0 authorization framework." RFC 6749, 2012.
- [145] L. Wang, X. Deng, J. Gui, P. Jiang, F. Zeng, and S. Wan, "A review of urban air mobility-enabled intelligent transportation systems: Mechanisms, applications and challenges," *Journal of Systems Architecture*, p. 102902, 2023.
- [146] A. Bauranov and J. Rakas, "Designing airspace for urban air mobility: A review of concepts and approaches," *Progress in Aerospace Sciences*, vol. 125, p. 100726, 2021.
- [147] U. Pelli and R. Riedel, "Flying-cab drivers wanted," *McKinsey Center for Future Mobility*, 2020.
- [148] S. P. Bharati, Y. Wu, Y. Sui, C. Padgett, and G. Wang, "Real-time obstacle detection and tracking for sense-and-avoid mechanism in uavs," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 2, pp. 185–197, 2018.
- [149] J. James, J. J. Ford, and T. L. Molloy, "Learning to detect aircraft for long-range vision-based sense-and-avoid systems," *IEEE Robotics and Automation Letters*, vol. 3, pp. 4383–4390, 2018.
- [150] R. Opromolla and G. Fasano, "Visual-based obstacle detection and tracking, and conflict detection for small uas sense and avoid," *Aerospace Science and Technology*, vol. 119, p. 107167, 2021.
- [151] Z. W. Lee, W. H. Chin, and H. W. Ho, "Air-to-air micro air vehicle interceptor with an embedded mechanism and deep learning,"

- Aerospace Science and Technology*, vol. 135, p. 108192, 2023.
- [152] K. R. Beier and H. Gemperlein, “Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation,” *Aerospace Science and Technology*, vol. 8, pp. 63–71, 2004.
- [153] O. O. Medaiyese, M. Ezuma, A. P. Lauf, and I. Guvenc, “Wavelet transform analytics for rf-based uav detection and identification system using machine learning,” *Pervasive and Mobile Computing*, vol. 82, p. 101569, 2022.
- [154] R. Sabatini, A. Gardi, and M. Richardson, “Lidar obstacle warning and avoidance system for unmanned aircraft,” *International Journal of Mechanical, Aerospace, Industrial and Mechatronics Engineering*, vol. 8, no. 4, pp. 718–729, 2014.
- [155] J. A. Paredes, F. J. Álvarez, M. E. Hansard, and K. Z. Rajab, “A gaussian process model for uav localization using millimetre wave radar,” *Expert Syst. Appl.*, vol. 185, p. 115563, 2021.
- [156] M. Champion, P. Ranganathan, and S. Faruque, “A review and future directions of uav swarm communication architectures,” in *2018 IEEE international conference on electro/information technology (EIT)*, pp. 0903–0908, IEEE, 2018.
- [157] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, “Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1020–1027, 2021.
- [158] “Airborne object tracking dataset.” <https://registry.opendata.aws/airborne-object-tracking>.  
Accessed: 2023-07-23.
- [159] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, “Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11682–11692, 2020.
- [160] M. Tremblay, S. S. Halder, R. De Charette, and J.-F. Lalonde, “Rain rendering for evaluating and improving robustness to bad weather,” *International Journal of Computer Vision*, vol. 129, pp. 341–360, 2021.
- [161] F. C. Akyon, S. O. Altinuc, and A. Temizel, “Slicing aided hyper



- inference and fine-tuning for small object detection,” in *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 966–970, IEEE, 2022.
- [162] S. Ghosh, J. Patrikar, B. Moon, M. M. Hamidi, and S. Scherer, “Air-track: Onboard deep learning framework for long-range aircraft detection and tracking,” 2023.
- [163] Y. Dong, C. Kang, J. Zhang, Z. Zhu, Y. Wang, X. Yang, H. Su, X. Wei, and J. Zhu, “Benchmarking robustness of 3d object detection to common corruptions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1022–1032, 2023.
- [164] S. Li, Z. Wang, F. Juefei-Xu, Q. Guo, X. Li, and L. Ma, “Common corruption robustness of point cloud detectors: Benchmark and enhancement,” *IEEE Transactions on Multimedia*, 2023.
- [165] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” *arXiv preprint arXiv:1903.12261*, 2019.
- [166] A. Khosravian, A. Amirkhani, H. Kashiani, and M. Masih-Tehrani, “Generalizing state-of-the-art object detectors for autonomous vehicles in unseen environments,” *Expert Systems with Applications*, vol. 183, p. 115417, 2021.
- [167] Y. Ghasemi, H. Jeong, S. H. Choi, K.-B. Park, and J. Y. Lee, “Deep learning-based object detection in augmented reality: A systematic review,” *Computers in Industry*, vol. 139, p. 103661, 2022.
- [168] A. Van Etten, “Satellite imagery multiscale rapid detection with windowed networks,” in *2019 IEEE winter conference on applications of computer vision (WACV)*, pp. 735–743, IEEE, 2019.
- [169] A. Arsenos, E. Petrongonas, O. Filippopoulos, C. Skliros, D. Kollias, and S. Kollias, “Nefeli: A deep-learning detection and tracking pipeline for enhancing autonomy in advanced air mobility,” *Available at SSRN 4674579*.
- [170] S. Wang, R. Veldhuis, and N. Strisciuglio, “The robustness of computer vision models against common corruptions: a survey,” *arXiv preprint arXiv:2305.06024*, 2023.
- [171] O. F. Kar, T. Yeo, A. Atanov, and A. Zamir, “3d common corruptions and data augmentation,” in *Proceedings of the IEEE/CVF Con-*

- ference on Computer Vision and Pattern Recognition*, pp. 18963–18974, 2022.
- [172] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, “Benchmarking robustness in object detection: Autonomous driving when winter is coming,” *arXiv preprint arXiv:1907.07484*, 2019.
- [173] D. Hendrycks and T. Dietterich, “Benchmarking neural network robustness to common corruptions and perturbations,” *arXiv preprint arXiv:1903.12261*, 2019.
- [174] C. Kamann and C. Rother, “Benchmarking the robustness of semantic segmentation models,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8828–8838, 2020.
- [175] J. Wang, S. Jin, W. Liu, W. Liu, C. Qian, and P. Luo, “When human pose estimation meets robustness: Adversarial algorithms and benchmarks,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11855–11864, 2021.
- [176] M. Chen, Z. Wang, and F. Zheng, “Benchmarks for corruption invariant person re-identification,” *arXiv preprint arXiv:2111.00880*, 2021.
- [177] H. He, J. Ding, and G.-S. Xia, “On the robustness of object detection models in aerial images,” *arXiv preprint arXiv:2308.15378*, 2023.
- [178] K. Konen and T. Hecking, “Increased robustness of object detection on aerial image datasets using simulated imagery,” in *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 1–8, IEEE, 2021.
- [179] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical poissonian-gaussian noise modeling and fitting for single-image raw-data,” *IEEE transactions on image processing*, vol. 17, no. 10, pp. 1737–1754, 2008.
- [180] G. Bradski, “The opencv library.,” *Dr. Dobb’s Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.
- [181] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, “Bot-sort: Robust associations multi-pedestrian tracking,” 2022.
- [182] Y. Du, Z. Zhao, Y. Song, Y. Zhao, F. Su, T. Gong, and H. Meng,

- “Strongsort: Make deepsort great again,” *IEEE Transactions on Multimedia*, 2023.
- [183] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, “Learning generalisable omni-scale representations for person re-identification,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5056–5069, 2021.
- [184] Y. Zhang, M. Li, R. Li, K. Jia, and L. Zhang, “Exact feature distribution matching for arbitrary style transfer and domain generalization,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8035–8045, 2022.
- [185] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, 2016.
- [186] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” in *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pp. 1–21, Springer, 2022.
- [187] Y. Du, J. Wan, Y. Zhao, B. Zhang, Z. Tong, and J. Dong, “Giao-tracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021,” in *Proceedings of the IEEE/CVF International conference on computer vision*, pp. 2809–2819, 2021.
- [188] D. Stadler and J. Beyerer, “Modelling ambiguous assignments for multi-person tracking in crowds,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 133–142, 2022.
- [189] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, “Ocean: Object-aware anchor-free tracking,” in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, pp. 771–787, Springer, 2020.
- [190] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgb-d images,” in *ECCV*, 2012.
- [191] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zit-

- nick, “Microsoft COCO: common objects in context,” *CoRR*, vol. abs/1405.0312, 2014.
- [192] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.
- [193] D. Organisciak, M. Poyser, A. Alsehaim, S. Hu, B. K. Isaac-Medina, T. P. Breckon, and H. P. Shum, “Uav-reid: A benchmark on unmanned aerial vehicle re-identification in video imagery,” in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2022.
- [194] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13001–13008, 2020.
- [195] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, “Hota: A higher order metric for evaluating multi-object tracking,” *International journal of computer vision*, vol. 129, pp. 548–578, 2021.
- [196] P. Dendorfer, A. Osep, A. Milan, K. Schindler, D. Cremers, I. Reid, S. Roth, and L. Leal-Taixé, “Motchallenge: A benchmark for single-camera multiple target tracking,” *International Journal of Computer Vision*, pp. 1–37, 2020.
- [197] Y. Ming, X. Meng, C. Fan, and H. Yu, “Deep learning for monocular depth estimation: A review,” *Neurocomputing*, vol. 438, pp. 14–33, 2021.
- [198] Y. Cao, Z. Wu, and C. Shen, “Estimating depth from monocular images as classification using deep fully convolutional residual networks,” 2017.
- [199] G. Fasano, D. Accado, A. Moccia, and D. Moroney, “Sense and avoid for unmanned aircraft systems,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 31, no. 11, pp. 82–110, 2016.
- [200] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” 2016.
- [201] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and

- J. W. Vaughan, “A theory of learning from different domains,” *Machine Learning*, vol. 79, pp. 151–175, 2010.
- [202] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” 2015.
- [203] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.
- [204] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012.
- [205] G. Jocher, A. Stoken, J. Borovec, NanoCode012, ChristopherSTAN, L. Changyu, Laughing, tkianai, A. Hogan, lorenzomamma, yxNONG, AlexWang1900, L. Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, F. Ingham, Frederik, Guilhen, Hatovix, J. Poznanski, J. Fang, L. Y. , changyu98, M. Wang, N. Gupta, O. Akhtar, PetrDvoracek, and P. Rai, “ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements,” Oct. 2020.
- [206] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLO,” Jan. 2023.
- [207] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “Yolox: Exceeding yolo series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.
- [208] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” 2018.
- [209] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.
- [210] S. Chen, P. Sun, Y. Song, and P. Luo, “Diffusiondet: Diffusion model for object detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19830–19843, 2023.
- [211] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” in *European conference on computer vision*, pp. 213–229, Springer, 2020.
- [212] X. Zhou, V. Koltun, and P. Krähenbühl, “Probabilistic two-stage detection,” in *arXiv preprint arXiv:2103.07461*, 2021.
- [213] M. C. Keles, B. Salmanoglu, M. S. Guzel, B. GURSOY, and G. E. Bostanci, “Evaluation of yolo models with sliced inference for small object detection,” *arXiv preprint arXiv:2203.04799*, 2022.
- [214] K. Zhou and T. Xiang, “Torchreid: A library for deep learning per-

- son re-identification in pytorch,” *arXiv preprint arXiv:1910.10093*, 2019.
- [215] X. Chang, T. M. Hospedales, and T. Xiang, “Multi-level factorisation net for person re-identification,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2109–2118, 2018.
- [216] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- [217] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.
- [218] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, “Observation-centric sort: Rethinking sort for robust multi-object tracking,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9686–9696, 2023.
- [219] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [220] A. Archet, N. Gac, F. Orieux, and N. Ventroux, “Embedded AI performances of Nvidia’s Jetson Orin SoC series,” in *17ème Colloque National du GDR SOC2*, (Lyon, France), June 2023.
- [221] E. Petrongonas, V. Leon, G. Lentaris, and D. Soudris, “Paralos: A scheduling memory management framework for heterogeneous vpus,” in *2021 24th Euromicro Conference on Digital System Design (DSD)*, pp. 221–228, 2021.
- [222] A. Kyriakos, E.-A. Papatheofanous, B. Charalampos, E. Petrongonas, D. Soudris, and D. Reisis, “Design and performance comparison of cnn accelerators based on the intel movidius myriad2 soc and fpga embedded prototype,” in *2019 International Conference on Control, Artificial Intelligence, Robotics Optimization (ICCAIRO)*, pp. 142–147, 2019.
- [223] A. K. A. Al Ghadani, W. Mateen, and R. G. Ramaswamy, “Tensor-based cuda optimization for ann inferencing using parallel acceleration on embedded gpu,” in *Artificial Intelligence Applications and Innovations* (I. Maglogiannis, L. Iliadis, and E. Pimenidis, eds.), (Cham), pp. 291–302, Springer International Publishing, 2020.

- [224] O. Shafi, C. Rai, R. Sen, and G. Ananthanarayanan, “Demystifying tensorrt: Characterizing neural network inference engine on nvidia edge devices,” in *2021 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 226–237, 2021.
- [225] “Ardupilot.” <https://ardupilot.org/>, 2023.  
[Online; accessed 7-November-2023].

