Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Robustness of Object Detectors against Corrupted Inputs

Diploma Thesis

by

Evangelia Koskinioti

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π

Αθήνα, Μάρτιος 2023

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Πληροφορικής
Εργαστήριο Τεχνολογίας Πληροφορικής και Υπολογιστών

Robustness of Object Detectors against Corrupted Inputs

Diploma Thesis

by

Evangelia Koskinioti

Επιβλέπων: Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Μαρτίου, 2023.

.....	.....	.....
Γεώργιος Στάμου	Αθανάσιος Βουλόδημος	Στέφανος Κόλλιας
Καθηγητής Ε.Μ.Π.	Επ. Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2023

(Υπογραφή)

.....
Ευαγγελια Κοσκινιωτη
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η στιβαρότητα των μοντέλων Όρασης Υπολογιστών είναι ένα ιδιαίτερα κοινό ζήτημα στη βιβλιογραφία, ειδικότερα δεδομένου του αυξανόμενου ρόλου που αυτά αποκτούν σε ένα ευρύ φάσμα τομέων στη σύγχρονη κοινωνία. Προκειμένου να εμπιστευθούμε τις αποφάσεις τους πρέπει να εξασφαλίσουμε την αξιοπιστία τους όταν αυτά χρησιμοποιούνται σε πραγματικά σενάρια, καθώς και την διαφάνειά τους, έτσι ώστε να αποφευχθούν ολέθρια λάθη και να μπορούν να παράγονται χρήσιμες ερμηνείες σχετικά με τη λειτουργία τους. Ειδικότερα σε εφαρμογές που λειτουργούν σε εξωτερικούς χώρους, όπως τα αυτό-οδηγούμενα αυτοκίνητα ή τα συστήματα ζωντανής πλοήγησης, τα μοντέλα εντοπισμού αντικειμένων πρέπει να έχουν τη δυνατότητα να λειτουργούν αποτελεσματικά και συστηματικά ακόμα και κάτω από δυσμενείς συνθήκες όπως η βροχή, το χιόνι, κάτω από διαφορετικές συνθήκες φωτισμού, ή σε περιπτώσεις δυσλειτουργίας εξοπλισμού που εισάγουν θόρυβο ή θόλωση στην εικόνα. Το συγκεκριμένο πρόβλημα έχει μελετηθεί εκτενώς στη βιβλιογραφία, τόσο για τον Εντοπισμό Αντικειμένων όσο και για την Ταξινόμηση Εικόνων, ωστόσο με την εισροή νέων μοντέλων που επιτυγχάνουν ολοένα και υψηλότερες αποδόσεις, μια αναλυτική μελέτη είναι απαραίτητη.

Σε αυτή την εργασία προσεγγίζουμε το συγκεκριμένο ζήτημα αξιολογώντας τα πιο σύγχρονα συστήματα Εντοπισμού Αντικειμένων σε ένα σύνολο αλλοιωμένων εικόνων και στη συνέχεια ερμηνεύοντας την απόδοσή τους, αρχικά χρησιμοποιώντας ένα σύνολο νέων ειδικά κατασκευασμένων μετρικών που βασίζονται στην μετρική AP, και στη συνέχεια οπτικά και ποσοτικά χρησιμοποιώντας την τεχνική των Χαρτών Εξοχής. Παρουσιάζουμε 18 σύνολα δεδομένων, καθένα από τα οποία περιλαμβάνει αλλοιωμένες εκδοχές του συνόλου δεδομένων COCO. Κάθε σύνολο περιέχει πέντε υποσύνολα: τις αρχικές 5.000 εικόνες αλλοιωμένες με μια διαφορετικού τύπου αλλοίωση η οποία έχει εφαρμοστεί με ένα αυξανόμενο επίπεδο σφοδρότητας. Θα χρησιμοποιήσουμε αυτά τα σύνολα δεδομένων προκειμένου να αξιολογήσουμε τη στιβαρότητα των τελευταίων μοντέλων YOLO και του μοντέλου Mask R-CNN, αναλύωντας την πτώση στην απόδοσή τους για κάθε τύπο αλλοίωσης, καθώς γίνεται πιο έντονη. Αυτή η αξιολόγηση θα πραγματοποιηθεί χρησιμοποιώντας τη μετρική mAP και εισάγοντας ένα σύνολο νέων μετρικών που έχουν κατασκευαστεί ειδικά για τη δομή των πειραμάτων μας, κατά την οποία η ένταση της αλλοίωσης αυξάνεται σε επίπεδα. Στη συνέχεια θα εξάγουμε τους Χάρτες Εξοχής για ένα υποσύνολο αυτών των αλλοιώσεων, προκειμένου να αποκτήσουμε οπτική κατανόηση του τρόπου που αυτά τα μοντέλα λαμβάνουν αποφάσεις, και πως αυτός επηρεάζεται. Τέλος, προτείνουμε ένα σύνολο μετρικών με σκοπό να εισάγουμε ποσοτική σκοπιά στους κατά τα άλλα οπτικούς Χάρτες Εξοχής, και να παρατηρήσουμε πιο καθαρά τις αλλαγές τους ανάλογα με τον τύπο της αλλοίωσης και τα επίπεδα σφοδρότητας. Καταφέραμε να εξάγουμε κάποιες ενδιαφέρουσες ερμηνείες και υποθέσεις βάσει των αποτελεσμάτων των πειραμάτων μας, και ταυτόχρονα να συνεισφέρουμε με τη μορφή μερικών νέων αλλοιώσεων και μετρικών, αλλά και να προτείνουμε κάποιες μελλοντικές κατευθύνσεις για τη συνέχιση των προσπαθειών μας που φαίνονται υποσχόμενες.

Λέξεις Κλειδιά — Εντοπισμός Αντικειμένων, Ταξινόμηση Εικόνων, Στιβαρότητα, Αλλοιώσεις Εικόνων, Χάρτες Εξοχής, YOLO, R-CNN

Abstract

Robustness of Computer Vision models is an ever-present issue in the literature, especially given the increasing role these models are playing in a wide range of domains in modern society. If we are to trust their decisions we need to ensure their reliability when deployed in real world scenarios, as well as their transparency, in order to be able to detect and prevent errors and offer valuable explanations regarding their way of operating. Especially in applications that operate in the outside world, such as self-driving cars or live navigation systems, object detectors need to be able to perform consistently even under conditions of heavy rain, snow, in different lighting conditions, or in cases of equipment malfunction that introduce noise or blur in the image. This problem has been studied extensively both in Image Classification and Object Detection, however with the influx of new state of the art Object Detectors, an analytical evaluation in neccessary.

In this thesis we approach this issue by evaluating the most modern Object Detection models on a set of corrupted images and then by interpreting their performance first using a set of custom metrics based on AP score and then visually and quantitatively using the technique of Saliency Maps. We present 18 datasets, each containing different corrupted versions of the COCO validation set. Every dataset contains five subsets: the original 5.000 images corrupted with a different corruption applied with increasing severity over a range of five levels. We will be using this dataset to evaluate the robustness of the newest YOLO object detectors and the Mask R-CNN object detector, by analysing the drop in performance for each corruption, as it gets more severe. This evaluation will be performed using the mAP score metric and by proposing a new set of metrics that are more tailored to our experimentation framework of increasing the severity level of the corruption. Next, we will be extracting the Saliency Maps for a subset of these corruptions, in order to gain a visual understanding of the way the models' decision making process is affected, and lastly we propose a set of metrics to attempt to insert a quantitative aspect to the otherwise qualitative visual saliency maps, and observe their fluctuations along with the corruption types and severity levels. We are able to formulate some interesting hypotheses and interpretations based on the results of our experiments, while also contributing a few new corruptions and metrics, and also proposing some promising future steps to continue our efforts.

Keywords — Object Detection, Image Classification, Robustness, Image Corruptions, Saliency Maps, YOLO, R-CNN

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντά μου, κ. Γεώργιο Στάμου, για την ευκαιρία να ενταχθώ στο εργαστήριό του, να μάθω και να εξεχλιχθώ. Θα ήθελα επίσης να ευχαριστήσω την Μαρία Λυμπεραίου για την πολύτιμη καθοδήγηση, υπομονή και υποστήριξη που μου προσέφερε στο περιπετειώδες αυτό διάστημα της εκπόνησης αυτής της εργασίας.

Τέλος θα ήθελα να πω ένα τεράστιο ευχαριστώ στην αγαπημένη μου οικογένεια, τους γονείς μου και τον αδερφό μου, καθώς η αγάπη και η στήριξή τους ήταν περισσότερη από ότι θα μπορούσε να ζητήσει ποτέ κανείς. Το μεγαλύτερο όμως ευχαριστώ ίσως το οφείλω στον Διονύση, που υπήρξε βράχος για μένα από τα πρώτα βήματα του ακαδημαϊκού μου ταξιδιού μέχρι τα τελευταία, καθώς είμαι βέβαιη ότι δεν θα είχα καταφέρει να βρίσκομαι εδώ σήμερα χωρίς εκείνον. Γι'αυτό τον λόγο του αφιερώνω αυτή τη διπλωματική, καθώς του οφείλω τα πάντα, και του εύχομαι ευτυχία στο καινούργιο του ταξίδι.

Κοσκινιώτη Ευαγγελία, Μάρτιος 2023

Contents

List of Figures

List of Tables

Chapter 1

Εκτεταμένη Περίληψη στα Ελληνικά

1.1 Θεωρητικό Υπόβαθρο

1.1.1 Όραση Υπολογιστών

Ο ρόλος της Τεχνητής Νοημοσύνης στη σύγχρονη κοινωνία είναι πλέον αδιαμφισβήτητος και ολοένα αυξανόμενος σε ένα εντυπωσιακό εύρος κλάδων, από γενικούς τομείς όπως η συλλογιστική και η αντίληψη, μέχρι συγκεκριμένες εφαρμογές στην Ιατρική, τη Νομική ή την αυτόνομη οδήγηση. Η διείσδυση των μοντέλων Τεχνητής Νοημοσύνης, και συγκεκριμένα Μηχανικής Μάθησης, στη λειτουργία και τη διαδικασία λήψης αποφάσεων αυτών των κλάδων προσφέρει άπλετα οφέλη όπως αυξημένες ταχύτητες, αυτοματοποίηση, ακρίβεια και μειωμένο κόστος σε διάφορες διαδικασίες, ωστόσο η αυξανόμενης σημασίας αποφάσεις που αναλαμβάνουν τα λάβουν εγείρουν ζητήματα ασφάλειας και στιβαρότητας. Για όλους τους παραπάνω λόγους, τα μοντέλα Μηχανικής Μάθησης δεν μπορούν να λειτουργούν πλέον σαν μαύρα κουτιά, αλλά είναι επιτακτική η ανάγκη εισαγωγής διαφάνειας στη λειτουργία τους προκειμένου να είναι δυνατός ο έλεγχος των αποτελεσμάτων τους προς αποφυγή λαθών και προς αύξηση της εμπιστοσύνης του κοινού προς αυτά.

Η Όραση Υπολογιστών (Computer Vision) είναι ένας από τους σημαντικότερους τομείς της Τεχνητής Νοημοσύνης που στοχεύει στην προσομοίωση του ανθρώπινου συστήματος όρασης και κατανόησης σε υπολογιστικά συστήματα. Στον πυρήνα της βρίσκεται η ανάπτυξη και η ανάλυση αλγορίθμων που μπορούν να αναλύσουν, να επεξεργαστούν και να ερμηνεύσουν ψηφιακές εικόνες και ακολουθίες ψηφιακών εικόνων και να εξάγουν ουσιώδη συμπεράσματα για αυτές. Οι εφαρμογές της Όρασης Υπολογιστών καλύπτουν ένα ευρύ φάσμα τομέων, από την Ιατρική όπου αλγόριθμοι καλούνται να αναλύσουν εικόνες ιατρικής απεικόνισης και να εντοπίσουν πιθανές ασθένειες σε αυτά (προσθετική όραση), έως την αυτόνομη οδήγηση, όπου αλγόριθμοι είναι υπεύθυνοι για τον εντοπισμό και την αναγνώριση αντικειμένων στον δρόμο, όπως πεζούς, σήματα ή άλλα οχήματα. Βάσει αυτών των παραδειγμάτων γίνεται προφανής η ύψιστη σημασία της κατοχύρωσης της ασφαλούς λειτουργίας αυτών των μοντέλων.

Από αυτά τα ζητήματα έχει εκκολαφθεί ο τομέας της Επεξηγήσιμης Τεχνητής Νοημοσύνης (Explainable AI, ή αλλιώς XAI), ο οποίος εστιάζει στη δημιουργία έμπιστων και διάφανων μοντέλων που μπορούν να ερμηνευθούν από ανθρώπους, παρέχοντας εξηγήσεις για τον τρόπο που λειτουργούν και τις αποφάσεις που λαμβάνουν. Σε αυτή την κατηγορία εντάσσεται, μεταξύ άλλων, η μελέτη της στιβαρότητας (robustness) των μοντέλων Μηχανικής Μάθησης, η οποία αφορά την αξιολόγηση της απόδοσης και της αξιοπιστίας τους σε διαφορετικές συνθήκες λειτουργίας, ειδικότερα σε συνθήκες που δεν έχουν αντιμετωπίσει κατά την εκπαίδευσή τους. Η μελέτη της στιβαρότητας ενός μοντέλου μπορεί να βοηθήσει στην αναγνώριση αδυναμιών, προκαταλήψεων ή κενών στην εκπαίδευση του και να προσφέρει μια καθαρότερη ματιά στη διαδικασία λήψης αποφάσεών του. Υπάρχουν διάφοροι τρόποι με τους οποίους μελετάται η στιβαρότητα μοντέλων Μηχανικής Μάθησης τα τελυταία χρόνια, όπως οι Ανταγωνιστικές Επιθέσεις (Adversarial Attacks), η Ανάλυση Ευαισθησίας (Sensitivity Analysis), οι Μεταβολές Κατανομής (Distribution Shifts) κλπ.

Σκοπός της εργασίας αυτής είναι η διεξοδική μελέτη της στιβαρότητας μοντέλων Εντοπισμού Αντικειμένων (Object Detection models) απέναντι σε αλλοιωμένες εισόδους με τη χρήση διάφορων τεχνικών, προκειμένου να εντοπιστούν τυχόν αδυναμίες ή προκαταλήψεις στον τρόπο που λειτουργούν. Συγκεκριμένα, θα αξιολογηθούν οι πιο σύγχρονοι αλγόριθμοι εντοπισμού αντικειμένων από τις δύο βασικές κατηγορίες: ενός σταδίου και δύο σταδίων και στη συνέχεια τα αποτελέσματα αυτά θα αναλυθούν σε μεγαλύτερο βάθος και από διαφορετική σκοπιά με τη χρήση των Χαρτών Εξοχής (Saliency Maps).

1.1.2 Εντοπισμός Αντικειμένων

Ο Εντοπισμός Αντικειμένων είναι μια από τις βασικότερες εργασίες του τομέα της Όρασης Υπολογιστών και περιλαμβάνει τον εντοπισμό και την αναγνώριση ορισμένων κλάσεων αντικειμένων σε εικόνες και βίντεο, επομένως μπορεί να αναλυθεί σε αυτές τις δύο υπό-εργασίες: τον εντοπισμό της θέσης ενός αντικειμένου σε μια εικόνα μέσω του καθορισμού ενός περιγράμματος συντεταγμένων (bounding box), το οποίο φράσσει τα όρια στα οποία περιλαμβάνεται το αντικείμενο, και την ταξινόμησή του σε μία προκαθορισμένη κλάση αντικειμένου.

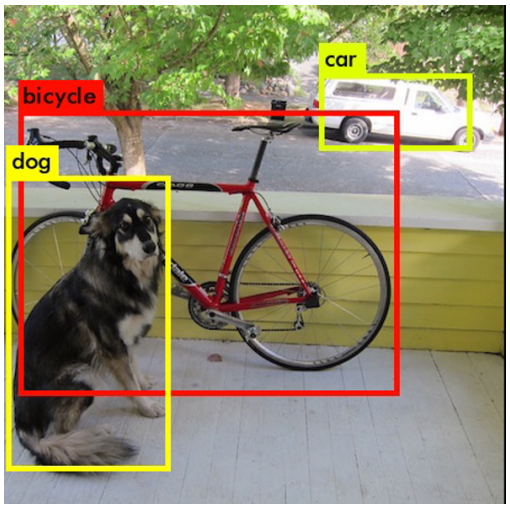


Figure 1.1.1: Εντοπισμός Αντικειμένων με Περιγράμματα Συντεταγμένων

Οι επικρατέστερες μέθοδοι που χρησιμοποιούνται για την επίλυση αυτού του προβλήματος ορίζουν ένα διαχωρισμό ανάμεσα στα μοντέλα που χρησιμοποιούνται, τα οποία μπορούν να ταξινομηθούν σε δύο κατηγορίες σύμφωνα με τον τρόπο που το προσεγγίζουν: τα μοντέλα ενός σταδίου (one-stage object detectors) και τα μοντέλα δύο σταδίων (two-stage object detectors). Τα δύο στάδια αυτά αναφέρονται στην παραδοσιακή προσέγγιση του Εντοπισμού Αντικειμένων όπου το μοντέλο επιτελούσε την εργασία σε δύο βασικά στάδια: την πρόταση περιοχών ενδιαφέροντος (region proposal stage) και την ταξινόμηση των αντικειμένων. Στο στάδιο της πρότασης περιοχών ενδιαφέροντος το μοντέλο παράγει ένα σύνολο υποψήφιων περιοχών (candidate regions), στις οποίες μπορεί να περιέχονται αντικείμενα, ενώ στο στάδιο της ταξινόμησης το μοντέλο κατατάσσει καθεμία από αυτές τις περιοχές σε μια κλάση αντικειμένου, εξάγοντας παράλληλα την πιθανότητα να περιέχεται ένα αντικείμενο στην περιοχή αυτή, και προσδιορίζει με μεγαλύτερη ακρίβεια την τοποθεσία του αντικειμένου.

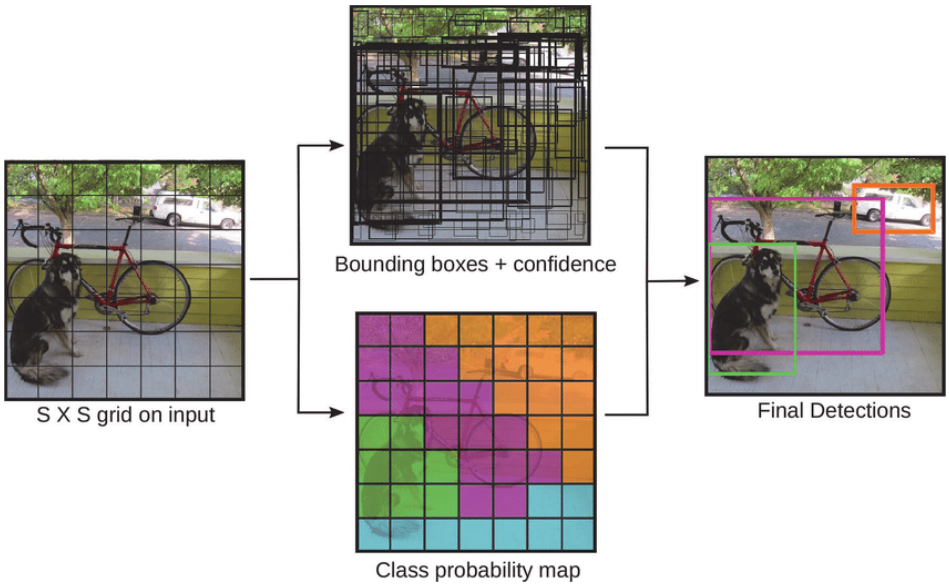


Figure 1.1.2: Διαδικασία Εντοπισμού Αντικειμένων σε Δύο Στάδια

Επομένως, τα μοντέλα που ανήκουν στην κατηγορία αυτή εντοπίζουν τα αντικείμενα μέσω αυτής της διαδικασίας, σε αντίθεση με τα μοντέλα ενός σταδίου, τα οποία προβλέπουν απευθείας τα περιγράμματα συντεταγμένων και τις πιθανότητες, οι οποίες ονομάζονται πιθανότητες κλάσης (class probabilities), σε ένα μόνο "πέρασμα", χωρίς δηλαδή το στάδιο της παραγωγής προτάσεων περιγραμμάτων συντεταγμένων. Τα περισσότερα σύγχρονα μοντέλα

μπορούν να ταξινομηθούν σε μία από αυτές τις δύο κατηγορίες, με τα πιο δημοφιλή μοντέλα που ανήκουν στην πρώτη κατηγορία να είναι τα μοντέλα της οικογένειας YOLO (You Only Look Once), που περιλαμβάνουν τουλάχιστον 8 εκδόσεις του αλγορίθμου με προοδευτικές βελτιώσεις και βελτιστοποίησεις [40], τα μοντέλα της οικογένειας EfficientDet [47], το μοντέλο SSD (Single Shot Detector) [25] και πολλά άλλα, ενώ στην δεύτερη κατηγορία κυριαρχεί η οικογένεια των R-CNN (Region-Based Convolutional Neural Networks) αλγορίθμων [14], που αποτελείται από μια ακολουθία μοντέλων που βασίζονται στην ίδια αρχή. Λόγω της βασικής διαφοράς στις αρχές λειτουργίας των μοντέλων που ανήκουν στις δύο κατηγορίες αυτές, καθεμία προσφέρει διαφορετικά οφέλη ανάλογα με την επιθυμητή χρήση: τα μοντέλα ενός σταδίου παρέχουν μεγαλύτερη ταχύτητα, καθώς δεν διαχωρίζουν τη διαδικασία εντοπισμού σε δύο στάδια, γεγονός που τα καθιστά ιδανικά για κινητές εφαρμογές, ενώ τα μοντέλα δύο σταδίων έχουν χαμηλότερη ταχύτητα προβλέψεων, προσφέρουν ωστόσο μεγαλύτερη ακρίβεια.

1.1.3 Χάρτες Εξοχής

Μια ακόμα έννοια που θα χρησιμοποιηθεί σε επερχόμενα κεφάλαια είναι η έννοια των Χαρτών Εξοχής, οι οποίοι αποτελούν μια τεχνική οπτικοποίησης που χρησιμοποιείται για να επισημάνει ποιες περιοχές και ποια στοιχεία είναι τα πιο σημαντικά, ή εξέχοντα, σε μια εικόνα. Η βασική τους χρησιμότητα έγκειται στην χρήση τους στη μελέτη στιβαρότητας μοντέλων Όρασης Υπολογιστών, όπου χρησιμοποιούνται προκειμένου να ερμηνεύσουν ποια χαρακτηριστικά της εικόνας είναι σημαντικότερα για το μοντέλο. Ένας χάρτης εξοχής είναι ένας χάρτης θερμότητας (heatmap), κάθε εικονοστοιχείο του οποίου περιέχει μια τιμή, η οποία αντιπροσωπεύει τη σημασία του συγκεκριμένου εικονοστοιχείου για το μοντέλο. Οι χάρτες εξοχής είναι κρίσιμης σημασίας εργαλεία στη μελέτη στιβαρότητας μοντέλων εντοπισμού αντικειμένων, και όχι μόνο, καθώς υπερβαίνουν τη βασική μέθοδο αξιολόγησης που βασίζεται σε μετρικές ακρίβειας, και επιτρέπουν την οπτική εξήγηση των προβλέψεων με τρόπο κοντινό στην ανθρώπινη σκέψη. Έτσι, έχουν τη δυνατότητα να αναδείξουν ποια συνολικά χαρακτηριστικά είναι σημαντικότερα στον εντοπισμό διάφορων αντικειμένων (χρώμα, κομμάτι του αντικειμένου, κλπ), τη σημασία του περιβάλλοντος ενός αντικειμένου (context clues) καθώς και πιθανές προκαταλήψεις στον τρόπο λειτουργίας τους (εξέταση κάποιου συγκεκριμένου, ανακριβούς χαρακτηριστικού για την αναγνώριση ενός αντικειμένου).

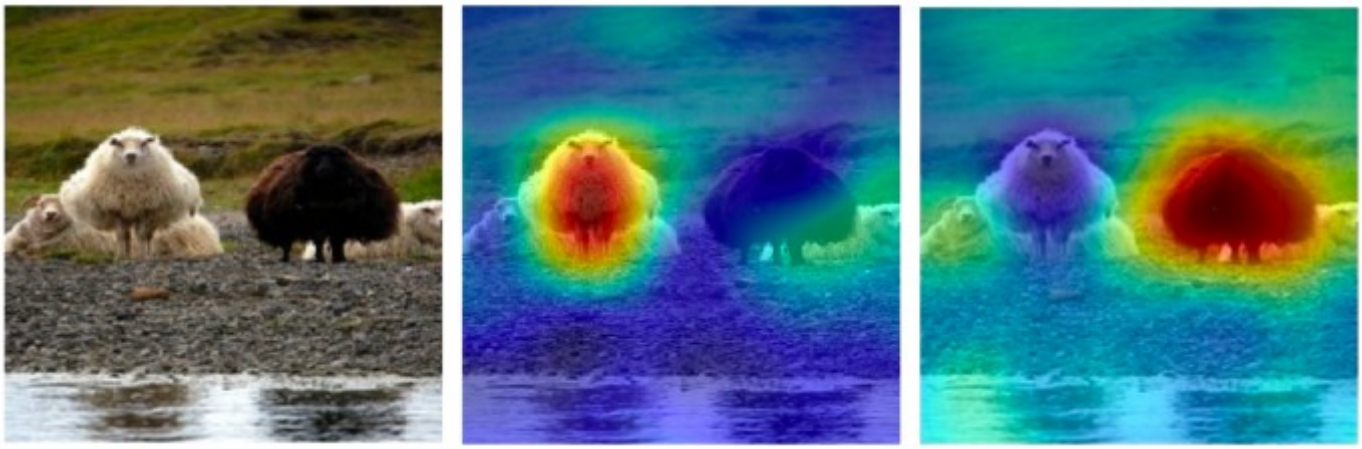


Figure 1.1.3: Παραδείγματα Χαρτών Εξοχής [34]

Ο καθορισμός του τι συνιστά μια "εξέχουσα" περιοχή σε μια εικόνα δεν είναι αυστηρός, επομένως η δημιουργία χαρτών εξοχής για μοντέλα εντοπισμού αντικειμένων προσεγγίζεται με πολλές διαφορετικές μεθόδους. Οι πιο κοινές μέθοδοι ονομάζονται gradient-based, και βασίζονται στον υπολογισμό του gradient της εξόδου ενός μοντέλου σε σχέση με την είσοδο, το οποίο χρησιμοποιείται τελικά σαν μέτρο για τη σημασία του κάθε εικονοστοιχείου της εισόδου σε σχέση με την έξοδο, και ο τελικός χάρτης συναρμολογείται από την συλλογή αυτών των μέτρων για όλα τα εικονοστοιχεία. Αυτές οι μέθοδοι είναι εύκολες στην υλοποίηση και εύκολα επεξηγήσιμες, γεγονός που δικαιολογεί το εύρος της χρήσης τους, ωστόσο απαιτούν γνώση της εσωτερικής αρχιτεκτονικής του μοντέλου, των βαρών και των νευρώνων του, προκειμένου να υπολογίσουν τα απαιτούμενα gradients. Ωστόσο, η πρόσβαση στην αρχιτεκτονική ενός μοντέλου δεν είναι πάντα εφικτή, και για αυτό τον λόγο αναπτύσσονται αλγόριθμοι "μαύρου κουτιού" για την ανάπτυξη χαρτών εξοχής. Αυτές οι μέθοδοι δεν απαιτούν γνώση των

παραμέτρων ενός μοντέλου, αλλά είναι γενικεύσιμες και μπορούν να εφαρμοστούν σε περισσότερα μοντέλα χωρίς τροποποίηση του αλγορίθμου για να προσαρμοστεί στην αρχιτεκτονική του μοντέλου. Μια τέτοια προσέγγιση, η οποία θα υιοθετηθεί και για τους πειραματισμούς της συγκεκριμένης εργασίας, είναι ο αλγόριθμος D-RISE [35]. Στα πλαίσια αυτής της μεθόδου, χρησιμοποιούνται δυαδικές μάσκες (binary masks) που καλύπτουν μέρος της εικόνας που δέχεται ως είσοδο το μοντέλο, και τα περιγράμματα συντεταγμένων που παράγονται από το μοντέλο με βάση αυτή την είσοδο συγκρίνονται με τα πραγματικά, τα οποία έχουν επισημανθεί κατά τη διάρκεια της κατασκευής του συνόλου δεδομένων. Με βάση αυτή τη διαφορά υπολογίζεται ένα βάρος για τη συγκεκριμένη δυαδική μάσκα και ο τελικός χάρτης εξοχής προκύπτει ως το άθροισμα των δυαδικών μασκώνω πολλαπλασιασμένων με τα αντίστοιχα βάρη. Η συγκεκριμένη μέθοδος φέρει το πλεονέκτημα της αγνωστικότητας απέναντι στον τύπο του μοντέλου, γεγονός που επιτρέπει την χρήση της για την εξήγηση των προβλέψεων διαφορετικών μοντέλων, τόσο ενός όσο και δύο σταδίων, με αποτέλεσμα να αποτελεί χρήσιμο εργαλείο στη μελέτη της στιβαρότητας και τη σύγκριση ανάμεσα στις δύο βασικές κατηγορίες μοντέλων εντοπισμού αντικειμένων.

1.1.4 Μελέτη Στιβαρότητας

Η στιβαρότητα ενός μοντέλου μηχανικής μάθησης αναφέρεται στην ικανοτήτά του να παράγει συστηματικά ακριβή αποτελέσματα υπό διαφορετικές συνθήκες. Στα πλαίσια της όρασης υπολογιστών και ειδικότερα του εντοπισμού αντικειμένων, η στιβαρότητα ενός μοντέλου μπορεί να οριστεί ως η ικανότητά του να εντοπίζει και να κατηγοριοποιεί αντικείμενα με ικανοποιητική ακρίβεια υπό την παρουσία θορύβων, μεταβαλλόμενων συνθηκών φωτισμού, επικάλυψης και άλλων παραγόντων που μπορούν να επηρεάσουν την ποιότητα της εικόνας εισόδου. Οι μέθοδοι που χρησιμοποιούνται για τη μελέτη της στιβαρότητας στην βιβλιογραφία είναι ποικίλες και μπορούν να διαχωριστούν ευρέως στις παρακάτω κατηγορίες:

- **Αλλοίωση Εικόνων Εισόδου**: Μια από τις συνηθέστερες προσεγγίσεις στη μελέτη στιβαρότητας είναι η αλλοίωση της εικόνας εισόδου με διάφορους τύπους διαταραχών, όπως θόρυβος, θόλωση, διαφορών ειδών καιρικές συνθήκες όπως χιόνι, βροχή κλπ, και η παρατήρηση της μείωσης της απόδοσης του μοντέλου με βάση τον τύπο της διαταραχής. Η συγκεκριμένη προσέγγιση προκύπτει φυσικά παρατηρώντας τις διάφορες εφαρμογές των αλγορίθμων όρασης υπολογιστών στην σύγχρονη καθημερινότητα: στην αυτόνομη οδήγηση όπου το μοντέλο που είναι υπεύθυνο να εντοπίζει αντικείμενα όπως πεζούς, άλλα οχήματα, σήματα και άλλα εμπόδια πρέπει να είναι ικανό να παράγει ακριβείς προβλέψεις ακόμα και κάτω από έντονα καιρικά φαινόμενα, σε ιατρικά βοηθητικά συστήματα καθοδήγησης ατόμων με προβλήματα όρασης, σε συστήματα ασφαλείας κ.ο.κ. Με τη μοντελοποίηση της απόδοσης υπό αυτές τις συνθήκες, σκοπός είναι η εξαγωγή μοτίβων και αιτιών για την πτώση της απόδοσης και η πρόταση μεθόδων για την επιδιόρθωση αυτής της αδυναμίας, είτε μέσω προ-εκπαίδευσης (pretraining) ή με χρήση διαφορετικών τεχνικών επαύξησης δεδομένων (data augmentation).
- **Επιθέσεις Αντιπαλότητας**: Ο όρος επιθέσεις αντιπαλότητας (adversarial attacks) αναφέρεται σε ένα σύνολο τεχνικών που χειραγωγούν τις εικόνες εισόδου ενός μοντέλου όρασης υπολογιστών με συγκεκριμένο τρόπο ώστε να "εξαπατήσουν" το μοντέλο, δηλαδή ώστε να εξάγει λανθασμένες προβλέψεις. Οι μέθοδοι αυτοί διαφέρουν από την απλή αλλοίωση εικόνων, καθώς συνήθως οι μεταβολές στις εικόνες είναι αδιόρατες για έναν ανθρώπινο παρατηρητή και είναι βελτιστοποιημένες ώστε να προκαλούν λανθασμένα αποτελέσματα από το μοντέλο. Οι επιθέσεις αντιπαλότητας θέτουν πολύ σοβαρά ζητήματα σχετικά με την ασφάλεια χρήσης μοντέλων μηχανικής μάθησης σε πραγματικά σενάρια, καθώς είναι δύσκολο να εντοπιστούν ακόμα και υπό ανθρώπινη επίβλεψη και μπορούν να προκαλέσουν καταστροφικές συνέπειες. Επιστρέφοντας στο παράδειγμα της αυτόνομης οδήγησης, μια επίθεση αντιπαλότητας μπορεί να εξαπατήσει το μοντέλο και να το οδηγήσει να θεωρήσει ένα κόκκινο φανάρι, πράσινο, με αποτέλεσμα τη δημιουργία ατυχημάτων. Επομένως, η στιβαρότητα των μοντέλων εντοπισμού αντικειμένων απέναντι σε επιθέσεις αντιπαλότητας αποτελεί ένα ζήτημα ύψιστης σημασίας, αλλά και αντικείμενο συνεχούς μελέτης τα τελευταία χρόνια. Προκειμένου να βελτιωθεί η στιβαρότητα ενός μοντέλου απέναντι σε επιθέσεις αντιπαλότητας, ερευνητές έχουν επιστρατεύσει διάφορες μεθόδους άμυνας, όπως η εκπαίδευση σε εικόνες που έχουν αλλοιωθεί ανταγωνιστικά (adversarially perturbed images) προκειμένου το μοντέλο να έχει την ικανότητα να παράγει ορθά αποτελέσματα ακόμα και μέσα από τις μεταβολές, η τυχαιοποίηση ορισμένων παραμέτρων του μοντέλου ώστε οι επιτιθέμενοι να μην αποκτούν εύκολα πρόσβαση στον τρόπο λειτουργίας του, η εκτενέστερη προεπεξεργασία της εισόδου προκειμένου να αφαιρεθούν τυχόν αλλοιώσεις στην εικόνα, με τη χρήση φίλτρων αφαίρεσης θορύβου κλπ.
- **Μεταβολές Κατανομής**: Στον πλαίσια της μελέτης στιβαρότητας για μοντέλα όρασης υπολογιστών οι

μεταβολές κατανομής (distribution shifts) συνιστούν ένα ενδιαφέρον πεδίο, το οποίο αφορά στην μεταβολή της υποκείμενης κατανομής των δεδομένων στα οποία εκπαιδεύτηκε ένα μοντέλο κατά την πραγματική του χρήση. Αυτό το φαινόμενο μπορεί να αναφέρεται στην εξέταση ενός αλγορίθμου εντοπισμού αντικειμένων σε εικόνες που περιλαμβάνουν σκηνές, αντικείμενα, συνθήκες φωτισμού, ακόμα και οπτικές γωνίες, οι οποίες δεν βρίσκονται στο σύνολο δεδομένων πάνω στο οποίο εκπαιδεύτηκε. Ειδικότερα στον τομέα της όρασης υπολογιστών, όπου τα μοντέλα χρησιμοποιούνται κατ'εξοχήν σε πραγματικές συνθήκες όπου η κατανομή των δεδομένων δεν εγγυάται να είναι πανομοιότυπη με αυτή των δεδομένων εκπαίδευσης, η αξιοπιστία τους έγκειται σε μεγάλο βαθμό στην στιβαρότητά τους απέναντι σε μεταβολές κατανομής. Σύμφωνα με την βιβλιογραφία, η απόδοση υπό τέτοιες συνθήκες μπορεί να βελτιωθεί ξανά μέσω της επαύξησης δεδομένων, της χρήσης τεχνικών μεταφοράς γνώσης (transfer learning) και άλλων.

- **Χάρτες Εξοχής**: Οι χάρτες εξοχής είναι μια κοινή μέθοδος που αξιοποιείται συστηματικά κατά τη μελέτη στιβαρότητας μοντέλων όρασης υπολογιστών, καθώς όπως αναφέρθηκε παραπάνω, προσφέρουν μια μέθοδο οπτικοποίησης της "συλλογιστικής" που υπόκειται των αποφάσεών τους. Πέρα από τη χρήση τους στην ερμηνεία των προβλέψεων, την ανάδειξη συστηματικών προκαταλήψεων και τον εντοπισμό συστηματικών αδυναμιών με σκοπό την διόρθωσή τους, οι χάρτες εξοχής χρησιμοποιούνται για την παραγωγή επιθέσεων αντιπαλότητας οι οποίες μπορούν να αξιοποιηθούν στην ανάπτυξη συστημάτων άμυνας απέναντί τους, για τον εντοπισμό αδιόρατων επιθέσεων, αλλά και ως εργαλεία επαύξησης δεδομένων, καθώς μπορούν να παράγουν νέα δείγματα εκπαίδευσης με βάση τα σημαντικότερα χαρακτηριστικά της εικόνας.

1.2 Συνεισφορά

Στη συγκεκριμένη εργασία προτείνουμε ένα πλαίσιο αξιολόγησης της στιβαρότητας μιας σειράς μοντέλων Εντοπισμού Αντικειμένων ενός και δύο σταδίων απέναντι σε αλλοιωμένες εικόνες εισόδου, προκειμένου να διαπιστωθεί η πτώση της απόδοσής τους και πιθανώς να αποκαλυφθούν συγκεκριμένα μοτίβα ανάλογα με το είδος της αλλοίωσης της εικόνας.

Το πλαίσιο αυτό αποτελείται από την εφαρμογή των μοντέλων YOLOv5, YOLOv6, YOLOv7 και YOLOv8 και του μοντέλου Mask R-CNN σε ένα σύνολο 5000 εικόνων, οι οποίες αποτελούν το σύνολο επαλήθευσης του συνόλου δεδομένων MS COCO, στις οποίες έχει εφαρμοστεί ένα σύνολο 18 διαφορετικών αλλοιώσεων σε 5 αυξανόμενα επίπεδα έντασης. Στη συνέχεια, η απόδοση των μοντέλων θα αναλυθεί με βάση την ακρίβειά τους, και τα αποτελέσματα θα συγκριθούν τόσο ανάμεσα στις διαφορετικές εκδόσεις των YOLO μοντέλων, τόσο και ανάμεσα στα μοντέλα YOLO και το μοντέλο R-CNN, προκειμένου να εξακριβωθεί μια σύγκριση της διαφοράς της στιβαρότητας ανάμεσα στις δύο κατηγορίες μοντέλων Εντοπισμού Αντικειμένων. Τέλος, η ανάλυση θα συνεχιστεί με τη χρήση Χαρτών Εξοχής, με βάση τους οποίους θα επιχειρηθεί η εξακρίβωση των παραπάνω αποτελεσμάτων.

Σκοπός αυτής της διπλωματικής εργασίας είναι η διεύρυνση της μελέτης της στιβαρότητας των μοντέλων Μηχανικής Μάθησης στον τομέα της Όρασης Υπολογιστών και συγκεκριμένα στον Εντοπισμό Αντικειμένων, και η διεξοδική ανάλυση της συμπεριφοράς των πιο σύγχρονων μοντέλων απέναντι σε ένα εύρος συνθηκών πάνω στο οποίο πιθανώς να μην έχουν εκπαιδευτεί. Η συνεισφορά μας μπορεί να συνοψιστεί στα παρακάτω σημεία:

- Παρουσιάζουμε μια συστηματική μελέτη των πιο σύγχρονων αλγορίθμων YOLO απέναντι σε ένα ευρύ φάσμα αλλοιώσεων η οποία, από όσο γνωρίζουμε, απουσιάζει από την βιβλιογραφία. Η μελέτη μας περιέχει αναλυτικά αποτελέσματα για δυο διαφορετικές εκδόσεις των τεσσάρων πιο σύγχρονων μοντέλων YOLO, καθεμία από τις οποίες εξυπηρετεί διαφορετικούς σκοπούς και μπορεί να χρησιμοποιηθεί σε διαφορετικές εφαρμογές.
- Παρουσιάζουμε μια εμπεριστατωμένη σύγκριση της στιβαρότητας ανάμεσα στις δύο βασικότερες κατηγορίες μοντέλων εντοπισμού αντικειμένων απέναντι στις ίδιες αλλοιώσεις, προκειμένου να συσχετίσουμε τις διαφορετικές αρχιτεκτονικές αυτών των μοντέλων με τις συμπεριφορές τους.
- Προτείνουμε ένα νέο πλαίσιο αξιολόγησης μοντέλων εντοπισμού αντικειμένων με τη χρήση χαρτών εξοχής με την πρόταση τριών νέων μετρικών, με σκοπό την εμπεριστάτωση των αποτελεσμάτων μας και την προαγωγή της ποσοτικής χρήσης οπτικών μεθόδων εξηγήσεων στο πρόβλημα της μελέτης στιβαρότητας.
- Παρέχουμε αναλυτικά αποτελέσματα για τη συμπεριφορά κάθε μοντέλου υπό τα αυξανόμενα επίπεδα σφοδρότητας των αλλοιώσεων καθώς και αντίστοιχες ερμηνείες για τα αποτελέσματα αυτά, προκειμένου να

μπορούν να χρησιμοποιηθούν κατά τη διαδικασία επιλογής μοντέλων στο πλαίσιο διαφορετικών προβλημάτων.

- Παρέχουμε 18 σύνολα δεδομένων που περιέχουν τις παραλλαγές του συνόλου δεδομένων επαλήθευσης του COCO, καθένα από τα οποία περιλαμβάνει τις αρχικές 5000 εικόνες αλλοιωμένες με μία από τις 15 αλλοιώσεις που χρησιμοποιούνται στη βιβλιογραφία συμπληρωμένες από τις 3 νέες δικές μας αλλοιώσεις, σε 5 επίπεδα αυξανόμενης έντασης. Αυτά τα σύνολα δεδομένων είναι δημόσια διαθέσιμα και μποορύν να χρησιμοποιηθούν για διεύρυνση της συγκεκριμένης μελέτης, αλλά και άλλων προσπαθειών προς τη βελτίωση της στιβαρότητας μοντέλων εντοπισμού αντικειμένων.

1.3 Πειραματικό Μέρος

1.3.1 Μέθοδος

Η μέθοδος που θα ακολουθήσουμε κατά τη διάρκεια των πειραματισμών μας μπορεί να συνοψιστεί στα παρακάτω βήματα:

1. Δημιουργία αλλοιωμένων συνόλων δεδομένων πάνω στα οποία θα αξιολογηθούν τα μοντέλα
2. Εξαγωγή προβλέψεων των μοντέλων πάνω στα αλλοιωμένα δεδομένα
3. Αξιολόγηση των προβλέψεων βάσει προκαθορισμένων μετρικών
4. Εξαγωγή χαρτών εξοχής βάσει των προβλέψεων για επιλεγμένες αλλοιώσεις
5. Περαιτέρω ανάλυση αποτελεσμάτων χαρτών εξοχής βάσει νέων μετρικών

Στη συνέχεια θα επεξηγηθούν τα επιμέρους τμήματα αυτής της ακολουθίας.

1.3.2 Αλλοιώσεις Εικόνων

Πρώτο βήμα στους πειραματισμούς μας αποτελεί η αλλοίωση των δεδομένων πάνω στα οποία θα αξιολογηθούν τα μοντέλα με διάφορες μεταβολές που θα καλύπτουν ένα ευρύ φάσμα πιθανών σεναρίων. Ως βάση θα χρησιμοποιηθούν οι μεταβολές που παρουσιάστηκαν στο [18], σε πέντε επίπεδα αυξανόμενης σφοδρότητας. Μπορούν να διαχωριστούν σε κατηγορίες ανάλογα με το φαινόμενο το οποίο προσομοιώνουν ως εξής:

- **Θόρυβος**: Γκαουσιανός θόρυβος, Θόρυβος Βολής, Κρουστικός Θόρυβος
- **Θόλωση** : Θόλωση Απεστίασης, Θόλωση Γυαλιού, Θόλωση Κίνησης, Θόλωση Ζουμ
- **Καιρικά Φαινόμενα**: Χιόνι, Πάγος, Ομίχλη, Φωτεινότητα
- **Ψηφιακά Φαινόμενα**: Αντίθεση, Ελαστικός Μετασχηματισμός, Pixelation, Συμπίεση Jpeg

Στη συνέχεια προτείνουμε τρεις νέες αλλοιώσεις οι οποίες θα λειτουργήσουν συμπληρωματικά στα πειράματά μας.

- **Βροχή:** Η αλλοίωση αυτή είναι μια προφανής προσθήκη στις παραπάνω, καθώς αποτελεί βασικό καιρικό φαινόμενο το οποίο χρειάζεται να αντιμετωπιστεί από τις εφαρμογές του εντοπισμού αντικειμένων σε πραγματικά σενάρια. Εφαρμόζει βροχή στην εικόνα σε πέντε επίπεδα σφοδρότητας και βασίζεται στη βιβλιοθήκη imgaug.

Figure 1.3.1: Τα 5 Επίπεδα Σφοδρότητας για την αλλοίωση Βροχή

- **Σκοτάδι:** Ακόμα μια προφανης προσθήκη, η οποία σκοτεινιάζει προοδευτικά την εικόνα με σκοπό την προσομοίωση των συνθηκών της νύχτας, η οποία αποτελεί μια συνθήκη για την οποία οι εφαρμογές που αναφέρθηκαν παραπάνω πρέπει να είναι προετοιμασμένες.

Figure 1.3.2: Τα 5 Επίπεδα Σφοδρότητας για την αλλοίωση Σκοτάδι

- **Μάσκα:** Στα πλαίσια αυτής της αλλοίωσης ένα τυχαίο μέρος της εικόνας αποκρύπτεται με τη χρήση ορ-

θογωνικών μασκών. Για κάθε επίπεδο σφοδρότητας το ποσοστό της εικόνας που αποκρύπτεται αυξάνεται, με σκοπό να καθορίσουμε τη σημασία του περιβάλλοντος των αντικειμένων στον εντοπισμό τους αλλά και τη συμπεριφορά των μοντέλων όταν το συνηθισμένο περιβάλλον των αντικειμένων έχει αποκρυφθεί.

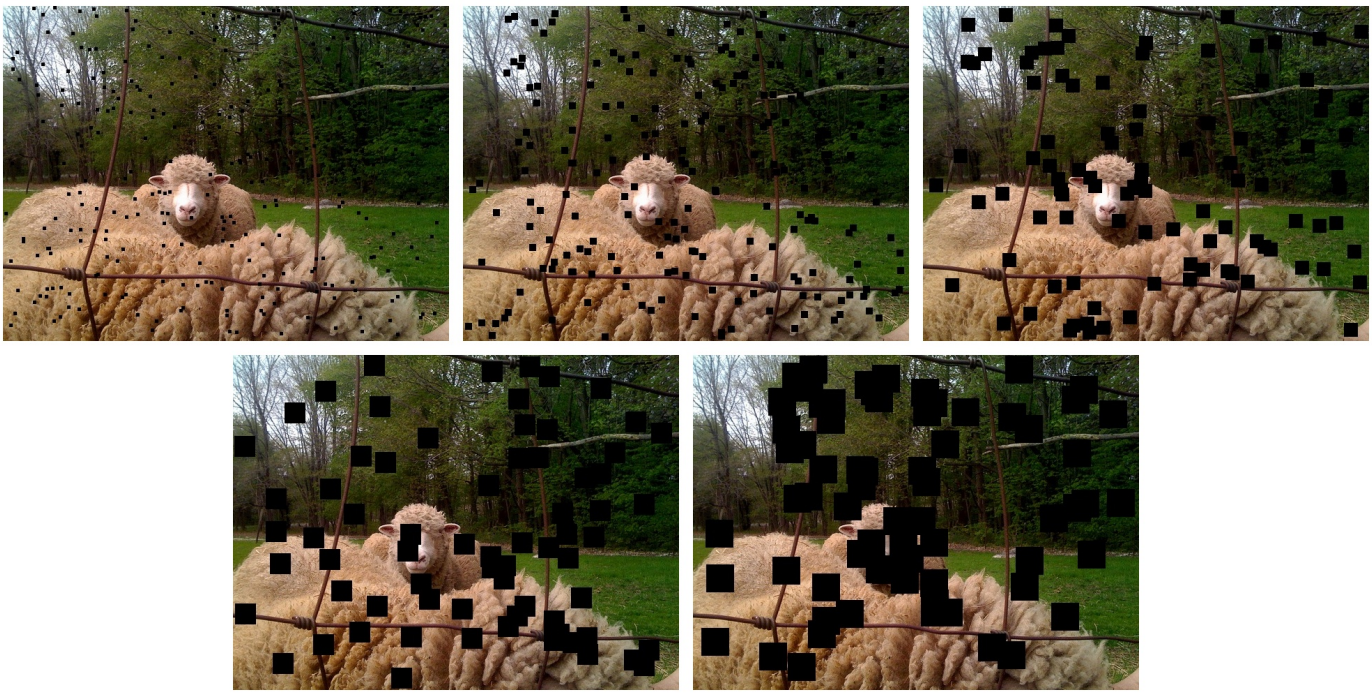


Figure 1.3.3: Τα 5 Επίπεδα Σφοδρότητας για την αλλοίωση Μάσκα

1.3.3 Σύνολο Δεδομένων και Μετρικές

Το σύνολο δεδομένων που θα αξιοποιηθεί στην παρούσα εργασία είναι το MS COCO (Common Objects in Conext), ίσως το πιο δημοφιλές σύνολο δεδομένων για την εργασία του εντοπισμού αντικειμένων, το οποίο αποτελείται από 330 χιλιάδες εικόνες 80 κλάσεων καθημερινών αντικειμένων στο σύνηθες περιβάλλον τους. Περισσότερες από 200 χιλιάδες από αυτές τις εικόνες συνοδεύονται από ετικέτες για διαφορετικές εργασίες, ο εντοπισμός αντικειμένων, η κατάτμηση αντικειμένων (object segmentation), η εκτίμηση βασικών σημείων (keypoint estimation) και άλλες. Συγκεκριμένα, θα αξιοποιηθεί το σύνολο επαλήθευσης του COCO, καθώς περιλαμβάνει έναν ικανοποιητικό όγκο είκονων (5,000 εικόνες) με τις αντίστοιχες ετικέτες τους (σε αντίθεση με το σύνολο δεδομένων εξέτασης), αλλά όχι υπερβολικά μεγάλο ώστε να καθιστά την επεξεργασία του προβληματική. Επιπρόσθετα, πολλά σύγχρονα μοντέλα εντοπισμού αντικειμένων έχουν εκπαιδευτεί πάνω στο σύνολο δεδομένων εκπαίδευσής του, επομένως η αξιολόγησή τους στο ίδιο σύνολο δεν θα ήταν αντικειμενική.

Με βάση αυτό το σύνολο εικόνων θα δημιουργηθούν 18 παραλλαγές του, στις οποίες θα έχει εφαρμοστεί μια από τις παραπάνω αλλοιώσεις, με τις ετικέτες να παραμένουν ίδιες.

Για την αξιολόγηση των πειραμάτων **Εντοπισμού Αντικειμένων** θα χρησιμοποιηθεί η μετρική mAP (mean Average Precision), η οποία συνδυάζει δύο κοινές μετρικές, την ακρίβεια (precision) και την ανάκληση (recall), ο ορισμός των οποίων παραλείπεται, ενώ χρησμοποιεί και την μετρική IoU (Intersection over Union, ή Τομή προς Ένωση). Το μέτρο αυτό χρησιμοποιείται για να καθορίσει την ομοιότητα ανάμεσα στην πρόβλεψη ενός περιγράμματος συντεταγμένων και το πραγματικό περίγραμμα. Ορίζεται ως το πηλικό του εμβαδού της τομής του πραγματικού περιγράμματος και του προβλεπόμενου περιγράμματος προς το εμβαδό της ένωσής του.

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{1.3.1}$$

Τιμή του IoU ίση με 1 ισοδυναμεί με τέλεια επικάλυψη ανάμεσα στα δύο περιγράμματα, ενώ η τιμή 0 ισοδυναμεί με πλήρη αποξένωση, επομένως και με μια ανακριβή πρόβλεψη.

Βασισμένη στο IoU, η μετρική mAP υπολογίζεται αρχικά καθορίζοντας ένα κατώφλι για την τιμή του IoU, ώστε οι προβλέψεις με τιμές μεγαλύτερες από το κατώφλι να θεωρούνται ως πραγματικές θετικές προβλέψεις (true positive), ενώ οι προβλέψεις με μικρότερες τιμές να θεωρούνται ως ψευδείς θετικές (false positive). Στη συνέχεια, σχεδιάζεται η καμπύλη ακρίβειας-ανάκλησης (precision-recall curve) μεταβάλλοντας το κατώφλι IoU και τελικά η τιμή AP υπολογίζεται ως το εμβαδό της καμπύλης. Τέλος, η μετρική mAP υπολογίζεται ως η μέση τιμή των AP όλων των διαφορετικών κλάσεων αντικειμένων. Η mAP μετρική είναι κατάλληλη για τον σκοπό των πειραμάτων μας σχετικά με τον εντοπισμό αντικειμένων καθώς αποτελεί το συνηθέστερο μέτρο αξιολόγησης των μοντέλων που αναπτύσσονται για αυτή την εργασία.

Συμπληρωματικά ως προς τη μετρική mAP ορίζουμε δύο ακόμα μετρικές οι οποίες είναι ειδικότερα προσαρμοσμένες στο πρόβλημα που προσεγγίζουμε. Αρχικά, ορίζουμε τη μετρική GmAP, η οποία αντιστοιχεί στη μέση τιμή των διαφορών ανάμεσα στις τιμές mAP ενός μοντέλου μεταξύ δύο διαδοχικών επιπέδων έντασης της ίδιας αλλοίωσης, ή αλλιως:

$$GmAP = \frac{\sum_{i=0}^{3}(mAP_{sev=i+1} - mAP_{sev=i})}{4} \tag{1.3.2}$$

Σκοπός αυτής της μετρικής είναι η συνολική αξιολόγηση της στιβαρότητας ενός μοντέλου απέναντι σε μια συγκεκριμένη αλλοίωση. Μια υψηλότερη τιμή GmAP ισοδυναμεί με ταχύτερη μείωση της απόδοσης του μοντέλου, επομένως σε μειωμένη στιβαρότητα. Τέλος, για τη συνολική αξιολόγηση της στιβαρότητας ενός μοντέλου απέναντι σε όλο το σύνολο των αλλοιώσεων ορίζουμε την μετρική mGmAP, η οποία ισούται με τη μέση τιμή όλων των GmAP που προέκυψαν από όλες τις διαφορετικές αλλοιώσεις. Η μετρική αυτή θα βοηθήσει στη σύγκριση ανάμεσα στη στιβαρότητα όλων των μοντέλων, και όχι στην αυστηρά ανώτερη απόδοση.

$$mGmAP = \frac{\sum_{i=corruption} GmAP_i}{\#\ Corruptions} \tag{1.3.3}$$

Από τη μεριά των διαταραχών ορίζουμε τη μετρική CmAP, σκοπός της οποίας είναι να καθορίσει ποια διαταραχή προκαλεί τα χειρότερα αποτελέσματα και σε ποια μοντέλα. Την ορίζουμε ως:

$$CmAP = \frac{\sum_{i=0}^{N} GmAP_i}{N} \tag{1.3.4}$$

όπου $GmAP_i$ είναι η τιμή GmAP του μοντέλου i σε αυτή τη διαταραχή, για έναν συνολικό αριθμό N μοντέλων. Όπως και πριν, όσο μεγαλύτερη είναι η απόλυτη CmAP τιμή, τόσο χειρότερη επίδραση έχει η συγκεκριμένη διαταραχή στα μοντέλα για τα οποία υπολογίστηκε.

Όσον αφορά την αξιολόγηση των πειραμάτων **Χαρτών Εξοχής**, δεν καταφέραμε να εντοπίσουμε κάποια εμπεριστατωμένη μετρική που να συλλαμβάνει την απόδοση των προβλέψεων που οπτικοποιούνται με τη βοήθεια των χαρτών, επομένως ορίζουμε ένα σύνολο πειραματικών μετρικών που θα ποσοτικοποιήσουν τα αποτελέσματων των πειραμάτων μας.

- **Αριθμός Εξεχόντων Περιοχών:** Αυτή η μετρική αναφέρεται στον αριθμό των εικονοστοιχείων του χάρτη εξοχής που επισημαίνονται ως "εξέχοντα", δηλαδή σημαντικά για την απόφαση του μοντέλου. Ορίζουμε ότι ένα εικονοστοιχείο θεωρείται εξέχον όταν η τιμή του στον χάρτη εξοχής υπερβαίνει ένα προκαθορισμένο κατώφλι, το οποίο ορίζουμε ως:

$$threshold = E[saliency\ map] + \frac{E[saliency\ map] + max(saliency\ map)}{2} \tag{1.3.5}$$

όπου ως

$$E[saliency\ map] = \frac{\sum_{i=0}^{N} saliency\ map(i)}{N} \tag{1.3.6}$$

ορίζεται η μέση τιμή των τιμών όλων των εικονοστοιχείων ενός εξέχοντος χάρτη, και ως $max(saliency\ map)$ ορίζεται η μέγιστη τιμή του χάρτη. Αυτή η μετρική καθορίστηκε πειραματικά, μετά από παρατήρηση των χαρτών που προέκυπταν από τις προβλέψεις των μοντέλων.

- **Λόγος Εικονοστοιχείων:** Η συγκεκριμένη μετρική αναφέρεται στην αναλογία του αριθμού εικονοστοιχείων που βρίσκονται εκτός ενός περιγράμματος συντεταγμένων προς τον συνολικό αριθμό εξεχόντων εικονοστοιχείων.

$$pixel\ ratio = \frac{S_N - S_B}{S_N} \tag{1.3.7}$$

  Σκοπός αυτής της μετρικής είναι να ποσοτικοποιήσει τη σημασία της περιοχής εντός του περιγράμματος συντεταγμένων για την απόφαση του μοντέλου αλλά και σε τι βαθμό το περιβάλλον εκτός του περιγράμματος επηρεάζει αυτή την απόφαση. Όταν ο λόγος εικονοστοιχείων παίρνει μεγάλες τιμές μπορούμε να υποθέσουμε ότι το περιβάλλον εκτός του περιγράμματος παίζει μεγάλο ρόλο στην πρόβλεψη, ή ότι το μοντέλο θεωρεί στοιχεία της επιβαλλόμενης αλλοίωσης ως σημαντικό περιβάλλον για την πρόβλεψη.

- **Λόγος Περιγραμμάτων:** Η τελική μετρική που ορίζουμε είναι ο λόγος περιγραμμάτων, ο οποίος αντιστοιχεί στην αναλογία των εικονοστοιχείων που βρίσκονται μέσα σε ένα περίγραμμα συντεταγμένων και χαρακτηρίζονται ως εξέχοντα προς τον συνολικό αριθμό εικονοστοιχείων στο περίγραμμα.

$$box\ ratio = \frac{S_B}{B} \tag{1.3.8}$$

  Αυτός ο λόγος στοχεύει στον καθορισμό του ποσοστού του περιγράμματος που είναι σημαντικό για την απόφαση του μοντέλου, και πως αυτό μεταβάλλεται με τις διάφορες αλλοιώσεις.

Όπως αναφέρθηκε παραπάνω, οι μετρικές αυτές βρίσκονται ακόμα σε πειραματικό στάδιο καθώς δεν έχουν εμπεριστατωθεί στη βιβλιογραφία και καθώς δεν είναι αυστηρά καθορισμένη η ερμηνεία τους.

1.3.4 Τα Μοντέλα

Στο πλαίσιο των πειραμάτων μας θα συμμετέχουν τα πιο αντιπροσωπευτικά μοντέλα κάθε κατηγορίας ανάμεσα στα μοντέλα ενός σταδίου και τα μοντέλα δύο σταδίων, δηλαδή οι οικογένειες YOLO και R-CNN αντίστοιχα. Συγκεκριμένα, θα αξιολογηθούν τα μοντέλα **YOLOv5**, **YOLOv6**, **YOLOv7**, **YOLOv8** από την οικογένεια YOLO και το μοντέλο **Mask R-CNN** από την οικογένεια R-CNN. Για κάθε μοντέλο YOLO θα χρησιμοποιηθούν δύο παραλλαγές: η έκδοση με τον μικρότερο αριθμό παραμέτρων, ή αλλιώς το απλούστερο μοντέλο, το οποίο επιτυγχάνει και τον χαμηλότερο χρόνο πρόβλεψης, και η έκδοση με τον μεγαλύτερο αριθμό παραμέτρων, δηλαδή το πιο περίπλοκο μοντέλο, το οποίο επιτυγχάνει την καλύτερη ακρίβεια αλλά συνοδεύεται από πιο αργούς χρόνους πρόβλεψης. Ο λόγος πίσω από αυτή την επιλογή είναι η διαπίστωση του αν ένα πιο περίπλοκο μοντέλο (δηλαδή ένα μοντέλο με περισσότερες παραμέτρους) είναι πιο στιβαρό από ένα πανομοιότυπο απλούστερο μοντέλο, ακόμα και αν έχει υψηλότερη ακρίβεια. Τα μοντέλα αυτά επιλέχθηκαν καθώς σε κάποια στιγμή κατά τη συγγραφή αυτής της εργασίας ήταν τα πιο σύγχρονα από την αντίστοιχη οικογένειά τους. Τα μοντέλα YOLO, όντας πιο σύγχρονα, είναι περισσότερα σε πλήθος καθώς κυκλοφόρησαν στο διάστημα του τελευταίου χρόνου, ενώ η οικογένεια R-CNN δεν παράγει πλέον μοντέλα με αντίστοιχη συχνότητα. Ωστόσο, η σύγκριση ανάμεσά τους δεν θα γίνει όσον αφορά την απόλυτα καλύτερη απόδοση, αλλά όσον αφορά την καλύτερη αντοχή και στιβαρότητα απέναντι στις επιβαλλόμενες αλλοιώσεις.

1.3.5 Αποτελέσματα

1.3.5.1 Εντοπισμός Αντικειμένων

Αρχικά θα αναφέρουμε τα αποτελέσματα του κεφαλαίου του Εντοπισμού Αντικειμένων, παραθέτοντας ενδεικτικά έναν πίνακα που περιλαμβάνει τα αποτελέσματα για όλα τα μοντέλα και όλες τις διαταραχές για το χαμηλότερο επίπεδο σφοδρότητας, το επίπεδο 1. Οι πίνακες για τα υπόλοιπα 4 επίπεδα σφοδρότητας για όλα τα μοντέλα μπορούν να βρεθούν στο Παράρτημα στο τέλος αυτής της εργασίας. Παρουσιάζουμε επίσης μια γραφική αναπαράσταση των αποτελεσμάτων για όλες τις διαταραχές και όλα τα επίπεδα σφοδρότητας ενδεικτικά για το μοντέλο YOLOv5n, ενώ τα διαγράμματα για τα υπόλοιπα μοντέλα μπορούν επίσης να βρεθούν στο Παράρτημα.

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.252	0.267	0.399	0.246	0.226
Frost	0.297	0.308	0.438	0.292	0.262
Fog	0.32	0.332	0.453	0.315	0.279
Brightness	0.357	0.362	0.48	0.24	0.345
Darken	0.349	0.356	0.468	0.347	0.315
Rain	0.339	0.345	0.468	0.336	0.325
Gauss	0.27	0.302	0.413	0.259	0.264
Impulse	0.224	0.28	0.366	0.221	0.194
Shot	0.27	0.306	0.412	0.262	0.263
Defocus	0.305	0.305	0.415	0.302	0.254
Zoom	0.131	0.131	0.208	0.127	0.108
Motion	0.289	0.306	0.409	0.289	0.269
Jpeg	0.282	0.313	0.347	0.277	0.263
Contrast	0.317	0.329	0.454	0.312	0.279
Pixelate	0.266	0.342	0.376	0.307	0.262
Elastic	0.298	0.318	0.418	0.308	0.278
Mask	0.274	0.251	0.404	0.257	0.255

Table 1.1: mAP Scores για Μικρά Μοντέλα - Επίπεδο Σφοδρότητας 1

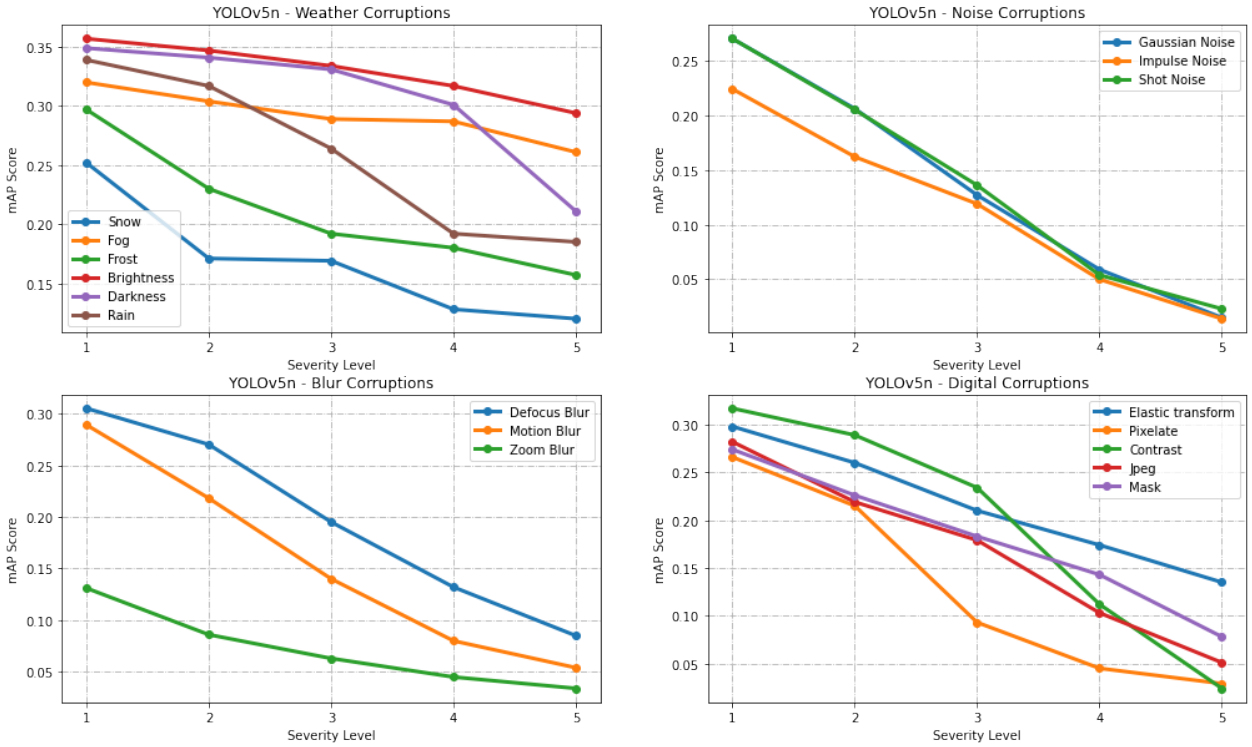


Figure 1.3.4: YOLOv5n απόδοση στο διαταραγμένο σύνολο δεδομένων COCO

Τέλος, παράγουμε και τα mGmAP αποτελέσματα για όλα τα μοντέλα προκειμένου να αποκτήσουμε μια συνολική εποπτεία της απόδοσής τους.

Detector	**mGmAP**
YOLOv5n	0.0601
YOLOv6n	0.0607
YOLOv7	0.067
YOLOv8n	0.06
Mask RCNN	**0.0583**
YOLOv5x	0.0713
YOLOv6l	0.065
YOLOv7E6	0.0681
YOLOv8x	0.06805

Table 1.2: Απόλυτες Αποδόσεις mGmAP

Υπάρχουν πολλές ενδιαφέρουσες παρατηρήσεις σχετικά με αυτά αποτελέσματα, τόσο για μεμονωμένα μοντέλα όσο και συνολικά. Καταρχάς, είναι ξεκάθαρο και αναμενόμενο ότι η απόδοση όλων των μοντέλων φθίνει καθώς η ένταση της διαταραχής αυξάνεται. Στη συνέχεια, όταν συγκρίνουμε τη στιβαρότητα μεταξύ των μοντέλων ενός και δύο σταδίων παρατηρούμε ότι το Mask R-CNN μοντέλο είναι πολύ πιο στιβαρό από όλες τις εκδόσεις των YOLO μοντέλων στις περισσότερες διαταραχές, παρόλο που είναι παλιότερο μοντέλο. Συγκεκριμένα, βλέπουμε ότι το Mask R-CNN έχει το χαμηλότερο mGmAP σκόρ από όλα τα μοντέλα, γεγονός που μας οδηγεί να υποθέσουμε ότι τα μοντέλα δύο σταδίων είναι πιο στιβαρά από τα μοντέλα ενός σταδίου, ωστόσο για να επιβεβαιωθεί μια τέτοια υπόθεση θα έπρεπε να πραγματοποιηθούν πιο εκτενή πειράματα με πιο σύγχρονα μοντέλα από τη μεριά των R-CNN.

Επίσης, μια πολύ ενδιαφέρουσα παρατήρηση είναι ότι ενώ τα μεγάλα μοντέλα YOLO έχουν συστηματικά ανώτερη απόδοση από τα μικρότερα αντίστοιχά τους, δεν ισχύει το ίδιο και για τις τιμές GmAP, με τα μικρά μοντέλα να είναι κατά μέσο όρο πιο στιβαρά. Με βάση αυτό το αποτέλεσμα μπορούμε να συμπεράνουμε ότι η ο μεγαλύτερος αριθμός παραμέτρων ενός δικτύου δεν ισοδυναμεί με αυξημένη στιβαρότητα, επομένως προκειμένου να βελτιωθούν τα μοντέλα σε αυτόν τον τομέα πρέπει το πρόβλημα να προσεγγιστεί με διαφορετικές μεθόδους από την αύξηση του μεγέθους και της πολυπλοκότητας και ίσως να χρειάζεται να χρησιμοποιηθούν τεχνικές σχετικές με την επαύξηση των δεδομένων εκπαίδευσης.

Συνεχίζουμε με τα αποτελέσματα της επίδρασης των διαταραχών σε όλα τα μοντέλα, μια μελέτη η οποία θα μας δείξει ποιες διαταραχές είναι οι πιο επικίνδυνες για αυτά, μεμονωμένα και συνολικά.

Corruption	**CmAP**
Snow	0.042
Frost	0.04
Fog	0.0173
Brightness	0.0208
Darken	0.0473
Rain	0.0383
Gauss	0.0938
Impulse	0.0807
Shot	0.0898
Defocus	0.796
Zoom	0.0412
Motion	0.09
Jpeg	0.084
Contrast	0.093
Pixelate	0.1
Elastic	0.0658
Mask	0.0665

Table 1.3: Absolute CmAP scores over all Detectors for all Corruptions

Παρατηρούμε ότι η διαταραχή Pixelate έχει συνολικά την χειρότερη επίδραση στο σύνολο των μοντέλων μας με τις Gaussian Noise, Contrast και Shot Noise να ακολουθούν. Μια πιθανή ερμηνεία που μπορούμε να δώσουμε σε αυτό το φαινόμενο είναι ότι το εφέ του pixelation μπορεί να οδηγήσει στην απώλεια πολλών λεπτομερειών μιας εικόνας και να κάνει τις ακμές της να χάνουν ανάλυση, γεγονός που οδηγεί στην απότομη πτώση της απόδοσης.

Συνολικά, μπορούμε επίσης να δούμε ότι οι διαταραχές του Θορύβου πέτυχαν μερικές από τις υψηλότερες τιμές CmAP, που σημαίνει ότι είχαν πολύ σημαντική επίδραση στα μοντέλα. Αυτό το αποτέλεσμα είναι ενδιαφέρον καθώς όταν προσθέτουμε θόρυβο σε μια εικόνα αλλάζουμε την υποκείμενη κατανομή της, γεγονός που έχει αποδειχθεί ότι επηρεάζει την απόδοση των μοντέλων σε πολύ μεγάλο βαθμό. Ωστόσο, ίσως το πιο σημαντικό μοτίβο που εμφανίζεται από τα αποτελέσματά μας είναι ότι οι διαταραχές που επηρεάζουν τις ακμές μιας εικόνας, κάνοντας την να φαίνεται πιο ομοιογενής, κάτι που κάνει δύσκολο τον διαχωρισμό ανάμεσα στα αντικείμενα και το φόντο, προκαλούν συνολικά τα χειρότερα αποτελέσματα. Αυτή η υπόθεση δικαιολογεί και την παρατήρηση ότι οι πιο λείες διαταραχές όπως Brightness, Fog, Darkness κλπ. δεν έχουν τόσο σημαντικές επιδράσεις.

Τέλος, επαναλαμβάνουμε αυτή την ανάλυση για κάθε μοντέλο, προκειμένου να έχουμε μια πιο λεπτομερή εικόνα για το ποιες διαταραχές επηρεάζουν ποια δίκτυα περισσότερο:

Detector	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**	**Mask RCNN**
Snow	0.044	0.0405	0.0425	0.0405	0.0225
Frost	0.042	0.04	0.0345	0.04	0.0235
Fog	0.0175	0.016	0.015	0.0175	0.012
Brightness	0.02	0.0165	0.02	0.02	0.0175
Darken	0.048	0.042	0.044	0.0455	0.0335
Rain	0.0405	0.035	0.029	0.0425	0.0255
Gauss	0.0945	0.0915	0.1025	0.0945	0.0395
Impulse	0.0825	0.083	0.089	0.0815	0.0275
Shot	0.0905	0.086	0.099	0.091	0.038
Defocus	0.08	0.078	0.0855	0.083	0.032
Zoom	0.0405	0.0415	0.048	0.0415	0.014
Motion	0.091	0.09	0.098	0.0895	0.0365
Jpeg	0.088	0.0795	0.0925	0.086	0.0355
Contrast	**0.103**	0.091	0.0895	0.091	**0.045**
Pixelate	0.1	**0.101**	**0.1165**	**0.103**	0.039
Elastic	0.065	0.0655	0.076	0.064	0.026
Mask	0.07	0.0715	0.0675	0.062	0.0285

Table 1.4: Απόλυτες τιμές CmAP για όλα τα Μοντέλα

Αποδομώντας τα αποτελέσματα του παραπάνω πίνακα συμπεραίνουμε ότι οι διαταραχές Pixelate και Contrast είναι αυτές που προκαλούν την μεγαλύτερη μείωση για όλα τα μοντέλα, ενώ και οι ακόλουθες διαταραχές είναι κυρίως κοινές ανάμεσα σε όλα τα μοντέλα. Αυτό το γεγονός μπορεί να είναι ένδειξη ότι η κατανομή των δεδομένων εκπαίδευσης και οι συνθήκες που περιλαμβάνει είναι πολύ σημαντικές για τη στιβαρότητα των μοντέλων, ίσως σημαντικότερες και από την ίδια την αρχιτεκτονική, καθώς διαφορετικά μοντέλα συμπεριφέρονται με τον ίδιο τρόπο.

1.3.5.2 Χάρτες Εξοχής

Παρουσιάζουμε τα αποτελέσματα των πειραματισμών μας με τους χάρτες εξοχής, ξεκινώντας από τη διαταραχή Contrast.

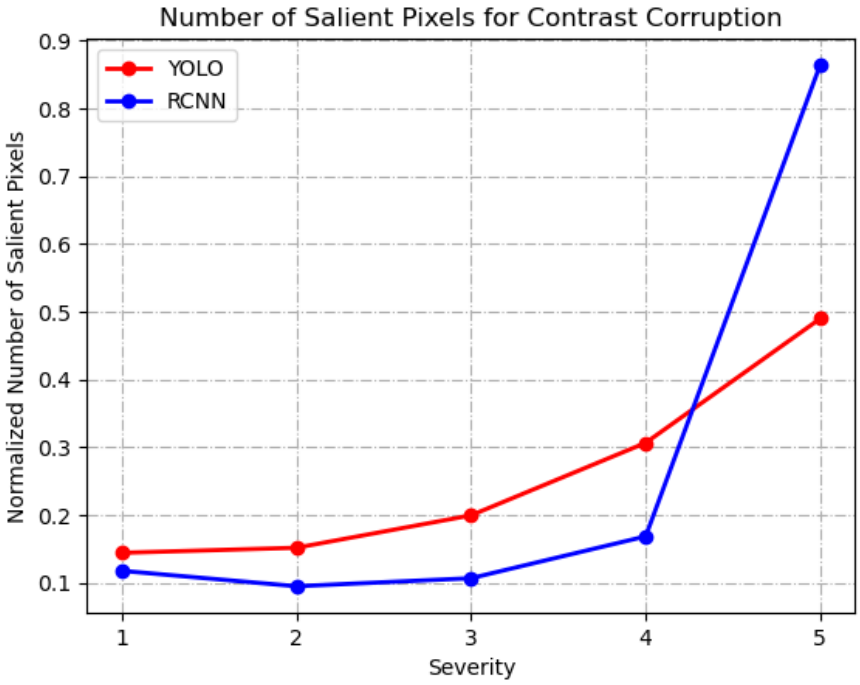


Figure 1.3.5: Αριθμός από εξέχοντα εικονοστοιχεία σε αλλοίωση Contrast (Contrast Corruption)

Όπως ήταν αναμενόμενο ο αριθμός των εξέχοντων εικονοστοιχείων αυξάνεται και για τα δύο μοντέλα ταυτόχρονα με την αύξηση της έντασης της διαραραχής με κάποιες μικρές διαφορές ανάμεσα στα δυο. Αυτό το φαινόμενο ήταν αναμενόμενο καθώς με την αύξηση της διαταραχής, το μοντέλο όχι μόνο εντοπίζει αντικείμενα που δεν υπάρχουν, αλλά λόγω της αλλαγής της υποκείμενης κατανομής των δεδομένων τα χαρακτηριστικά της εικόνας που θα θεωρούσε το μοντέλο φυσιολογικά πλέον θεωρούνται αποκλίσεις, δηλαδή κάτι στο οποίο πρέπει να δοθεί προσοχή.

Συνεχίζουμε με τον υπολογισμό των λόγων εικονοστοιχείων για τα μοντέλα μας υπό την ίδια διαταραχή ξεκινώντας με την κλάση Άτομο.

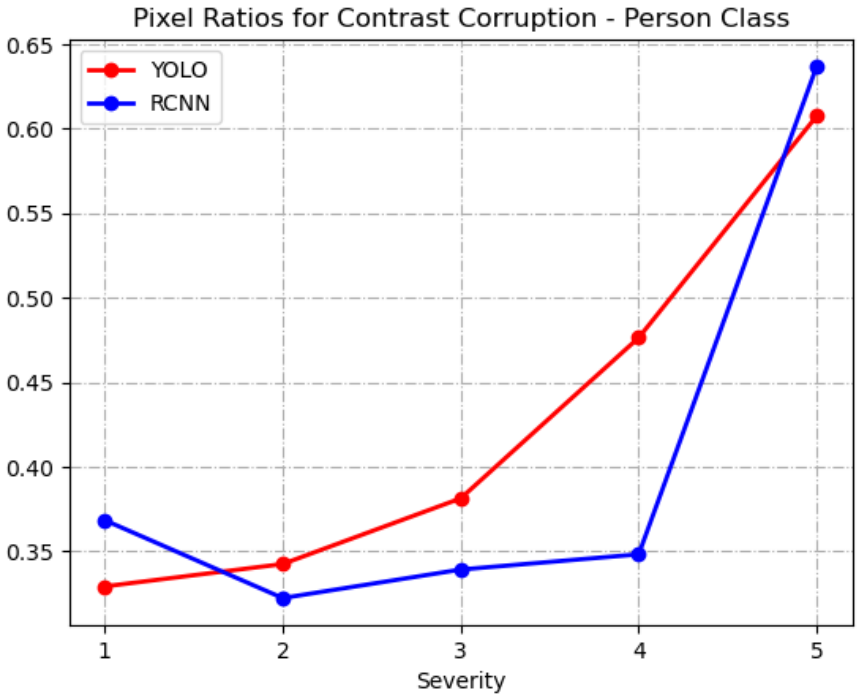


Figure 1.3.6: Λόγοι Εικονοστοιχείων στην αλλοίωση Contrast για την κλάση Άτομο (Person Class)

Παρατηρούμε ότι οι λόγοι εικονοστοιχείων αυξάνονται με την αύξηση της έντασης, γεγονός που σε συνδυασμό με την αύξηση των εξεχόντων εικονοστοιχείων μπορούμε με ασφάλεια να πούμε ότι το περιβάλλον της εικόνας αποκτά ολοένα μεγαλύτερο ενδιαφέρον, πιθανότατα για τους λόγους που αναφέραμε παραπάνω.

Τέλος υπολογίζουμε και τους λόγους περιγραμμάτων για την κλάση Άτομο για την ίδια διαταραχή.

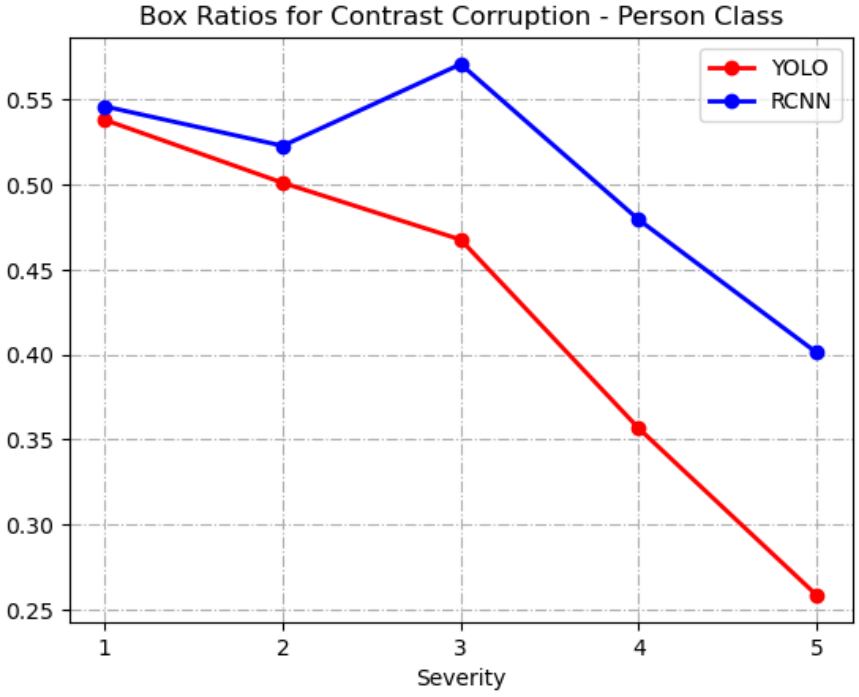


Figure 1.3.7: Λόγοι Περιγραμμάτων για την αλλοίωση Contrast - Κλάση Άτομο (Person Class)

Μπορούμε να δούμε ότι οι τιμές αυτής της μετρικής έχουν μια καθοδική τάση με την αύξηση της έντασης, γεγονός που σημαίνει ότι η περιοχή εντός του πλαισίου γίνεται ολοένα και λιγότερο σημαντική, γεγονός που σε συνδυασμό με τα παραπάνω συμπεράσματα μας οδηγεί στο συμπέρασμα ότι πράγματι το περιβάλλον του ίδιου του αντικειμένου γίνεται λιγότερο σημαντικό σε σχέση με το συνολικό περιβάλλον της εικόνας, ίσως για τους λόγους που αναφέραμε παραπάνω.

Θα επεκτείνουμε την ανάλυση μας στην κλάση Αυτοκίνητο η οποία είναι μια πολύ συχνά εμφανιζόμενη κλάση με ποικιλία εφαρμογών των μοντέλων που μελετάμε

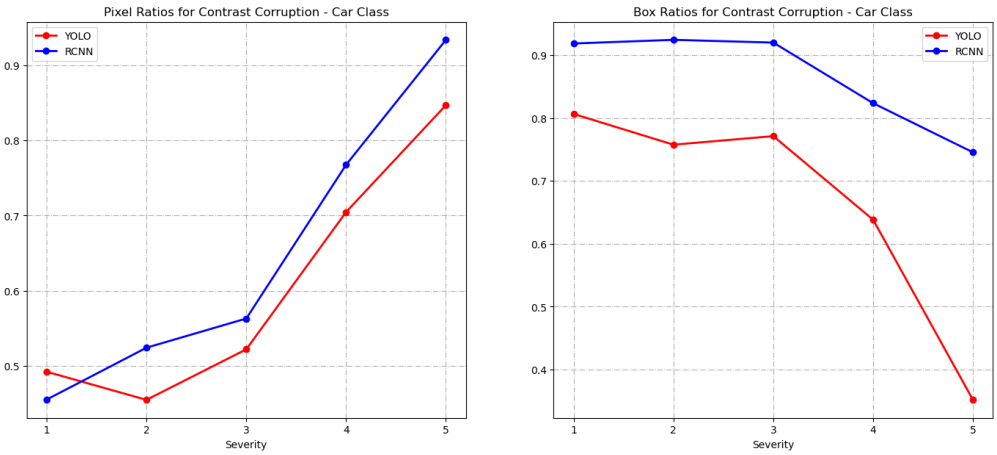


Figure 1.3.8: Εξέχουσες Μετρικές για την αλλοίωση Contrast - Κλάση Αυτοκίνητο

Όπως μπορούμε να παρατηρήσουμε το διάγραμμα έχει τις ίδιες γενικές αρχές με την κλάση Άτομο, παρότι τα αποτελέσματα δεν ταυτίζονται απόλυτα, όπως και είναι ανεμενόμενο. Οι τιμές του λόγου εικονοστοιχείων αυξάνονται μαζί με την αύξηση τον εξεχόντων εικονοστοιχείων, ενώ οι τιμές του λόγου περιγράμματος μειώνονται. Παρατηρήστε ότι ο αριθμός των εξεχόντων εικονοστοιχείων είναι μια μετρική ανεξάρτητη από την κλάση, δηλαδή αναφέρεται σε όλη την εικόνα και παραμένει ανεξάρτητη από το οποία κλάση ανιχνεύεται. Βασισμένοι σε αυτή την παρατήρηση μπορούμε να εικάσουμε ότι τα συμπεράσματα που βγάζουμε για μια κλάση μπορούν να γενικευτούν και σε άλλες.

Περισσότερες λεπτομέρειες για την ανάλυση των αποτελεσμάτων μπορούν να βρεθούν στο αγγλικό τμήμα αυτής της εργασίας, ωστόσο τα γενικότερα αποτελέσματα μπορούν να συσχετιστούν με αυτά που αναφέραμε παραπάνω, με κάποιες ειδικές εξαιρέσεις.

1.4 Σύνοψη, Συμπεράσματα και Μελλοντικές Κατευθύνσεις

1.4.1 Σύνοψη

Στα πλαίσια αυτής της διπλωματικής εργασίας πειραματιστήκαμε με τα πιο σύγχρονα μοντέλα Εντοπισμού Αντικειμένων και αναλύσαμε τη συμπεριφορά τους απέναντι σε εικόνες που έχουν αλλοιωθεί από ένα εύρος μεταβολών βάσει διαφόρων μεθόδων και μετρικών. Αρχικά, δημιουργήσαμε 18 σύνολα δεδομένων, καθένα από τα οποία περιλαμβάνει 25.000 εικόνες, τις 5.000 εικόνες του συνόλου δεδομένων επαλήθευσης του COCO αλλοιωμένες με 5 επίπεδα αυξανόμενης έντασης. Στη συνέχεια εφαρμόσαμε τα πιο σύγχρονα μοντέλα YOLO και R-CNN και καταγράψαμε την απόδοσή τους σε καθένα από αυτά τα σύνολα δεδομένων, για κάθε επίπεδο έντασης της κάθε αλλοίωσης. Έπειτα, εξάγαμε τους χάρτες εξοχής για τις αλλοιώσεις που είχαν τη σφοδρότερη επίδραση στα μοντέλα, και προχωρήσαμε σε συστηματική ποσοτική και ποιοτική ανάλυση των χαρτών προκειμένου να αποκτήσουμε περισσότερες λεπτομέρειες σχετικά με τη συμπεριφορά τους απέναντι σε αυτές τις αλλοιώσεις.

1.4.2 Συμπεράσματα

1.4.3 Μελλοντικές Κατευθύνσεις

Η έρευνα στο πεδίο της στιβαρότητας, και ειδικότερα στο πεδίο της όρασης υπολογιστών, περιλαμβάνει τρομερό ενδιαφέρον και αποτελεί ζήτημα ύψιστης σημασίας καθώς τα αυτά τα μοντέλα μηχανικής μάθησης εισέρχονται ολοένα και περισσότερο στις ζωές μας. Για αυτό τον σκοπό, αφορμώμενοι από την προσπάθειά μας προτείνουμε τα εξής μελλοντικά ερευνητικά βήματα:

- Την διεύρυνση του συνόλου αλλοιώσεων που εφαρμόζονται στις εικόνες ώστε να περιλάβει ένα ακόμα ευρύτερο φάσμα φαινομένων αλλά και τη βελτίωση των αλλοιώσεων που είναι ήδη διαθέσιμες με σκοπό την ακριβέστερη προσομοίωση των φαινομένων επομένως και την ακριβέστερη διαπίστωση της απόδοσης των μοντέλων. Ένα τέτοιο παράδειγμα αποτελεί η χρήση GANs για την παραγωγή των αλλοιώσεων και την καλύτερη προσομοίωση διαφορετικών συνθηκών, όπως ρεαλιστικές συνθήκες νύχτας, πραγματικές συνθήκες σκίασης, εφέ καπνού κ.α. Γενικότερα, όσο ευρύτερο είναι το φάσμα των αλλοιώσεων που μπορούμε να παρέχουμε στο μοντέλο, τόσο πιο περιεκτική και αναλυτική θα είναι η μελέτη που θα προκύψει, η οποία θα προάγει την παραγωγή πιο στιβαρών και αξιόπιστων μοντέλων.
- Την προσθήκη μιας εκτενούς μελέτης της στιβαρότητας αυτών και άλλων μοντέλων απέναντι σε εικόνες με σημαντική επικάλυψη μεταξύ αντικειμένων. Η επικάλυψη αποτελεί μια σημαντική πρόκληση όσον αφορά τον εντοπισμό αντικειμένων, καθώς προκαλεί σύγχυση στα μοντέλα και μειώνει την απόδοσή τους, ακόμα και υπό κανονικές συνθήκες. Επομένως, η ανάλυση της επίδρασης της επικάλυψης στα πιο σύγχρονα μοντέλα εντοπισμού αντικειμένων θα μπορούσε να φανεί ιδιαίτερα χρήσιμη στην ανάπτυξη τεχνικών που υπερβαίνουν αυτό το ζήτημα.
- Την επέκταση των πειραμάτων μας σε εισόδους της μορφής βίντεο. Ειδικά τα μοντέλα της οικογένειας YOLO φημίζονται για τις αποδόσεις τους σε βίντεο και χρησιμοποιούνται σε πολύ μεγάλο βαθμό για τον εντοπισμό αντικειμένων πραγματικού χρόνου. Επομένως, η παρατήρηση της απόδοσής τους απέναντι σε αλλοιωμένες ακολουθίες βίντεο υπόσχεται ενδιαφέροντα αποτελέσματα. Επιπρόσθετα, το φαινόμενο της επικάλυψης που αναφέρθηκε παραπάνω θα μπορούσε να εφαρμοστεί και στη συγκεκριμένη μελέτη της εισόδου βίντεο.
- Την επέκταση της ανάλυσης των χαρτών εξοχής σε ένα ευρύτερο φάσμα κλάσεων αντικειμένων. Όπως αναφέρθηκε στο αντίστοιχο κεφάλαιο, τα πειράματά μας περιλάμβαναν ένα υποσύνολο των κλάσεων του συνόλου δεδομένων COCO, οι οποίες διαθέτουν περισσότερη σημασία για ορισμένες εφαρμογές. Ωστόσο, δεδομένης της έλλειψης περιορισμών όσον αφορά τους διαθέσιμους υπολογιστικούς πόρους, αυτά τα πειράματα μπορούν να επαναληφθούν για όλες τις διαθέσιμες κλάσεις, προκειμένου να επαληθευθούν, και ίσως να γενικευθούν τα εξαχθέντα αποτελέσματα.
- Την υλοποίηση αυτού του πλαισίου πειραμάτων σε διαφορετικά μοντέλα, εκτός του τομέα της όρασης υπολογιστών. Μια ενδιαφέρουσα μελλοντική πιθανότητα προς αυτή την κατεύθυνση είναι ο πειραματισμός με μοντέλα Οπτικής Ερμηνείας Κοινής Λογικής (Visual Commonsense Reasoning models), τα οποία δεδομένης μιας οπτικής και μιας γραπτής εισόδου, μπορούν να παράγουν λογικά συμπεράσματα και ερμηνείες για μια σκηνή και τι συμβαίνει μέσα σε αυτή ή τι θα συμβεί στο άμεσο μέλλον. Η στιβαρότητα αυτών των μοντέλων παραμένει ένα σχετικά ανεξερεύνητο πεδίο, επομένως η εφαρμογή τους σε αλλοιωμένες εισόδους θα μπορούσε να επιφέρει μια σειρά ενδιαφέρον αποτελεσμάτων, ακόμα και για πιο βασικές αρχές της όρασης υπολογιστών.

Chapter 2

Introduction

2.1 Introduction

Artificial Intelligence (AI) is a general field of study that aims at the simulation of human intelligence in machines that are programmed to think and learn like humans. This interpretation of AI began as more of a fantasy, a wild fever dream of what computers might be able to achieve decades or even centuries into the future - perfectly simulate human intelligence. However, in recent years this dream seems to be edging ever closer to reality. The rapid advances in processing power and memory capabilities have brought forward the rise of Machine Learning (ML) models, which are essentially algorithms that rely on statistics and repetitive learning to improve their performance on a specific task whose applications are as impressive as they are widespread in several industries such as healthcare, finance, transportation, entertainment, security etc. The huge progress in this domain has been received with vibrant enthusiasm but also large amounts of concern and distrust on two major issues.

Firstly, it is known that ML models tend to underperform when presented with new, unexpected or adversarial inputs, or even inputs that were drawn from a distribution different than the one they were trained on. This is a major issue for many of the aforementioned applications, especially the ones that rely on the model's accurate performance under adverse conditions: for example you would still expect a self-driving car to safely get you to your destination even under heavy rain, snow, fog, or even when the input image gets somewhat corrupted due to camera issues. This is where Robustness study appears, which aims to create ML models that maintain their performance under unexpected inputs, whether those are corrupted inputs, inputs that come from a different distribution, adversarially manufactured inputs etc.

The second cause of concern around AI systems is that they mostly operate as black boxes: there is no clear justification behind the decisions they make, which puts their position in fields like Medicine or Law under question. How can we blindly trust the decisions these algorithms make when we do not know the reasoning behind them? These questions are the main foundation of the field of Explainable AI (XAI), which aims to create explanations (interpretations) for the models' predictions in order to prevent errors and create trustworthy, transparent systems. Robustness study plays an important role in model explainability, as a robust model is more likely to perform consistently and provide accurate explanations for its decisions.

Computer Vision (CV) is a field of Artificial Intelligence that focuses on enabling computers to interpret and understand visual data from the world around us. To that end, this field of study includes many relevant tasks such as Image Classification, Object Detection, Instance Segmentation, Pose Estimation etc., all aiming to encode a certain aspect of the human visual system. CV finds numerous applications the industries previously mentioned, such as creating diagnoses from medical images, detecting suspicious activity in security systems, autonomous driving etc, which is why robustness and explainability of CV models is crucial.

In this work, our aim is to systematically analyze the robustness of state of the art Object Detection models under corrupted inputs, to observe emerging patterns and provide meaningful interpretations for these models' performance. We will be solidifying this study using Saliency Maps, which are an explainability technique that enables visualization of an Image Classifier's of Object Detector's decisions by highlighting regions of the image that were important for the prediction. We will also be providing quantitative results for this primarily visual field, and attempting to correlate these results with our previous interpretations, while also creating new ones.

Our main contributions can be summarized in the points below:

- Creating a wide set of datasets that include images that have been corrupted with varying types of corruptions and levels of severity, some of which are already available in the literature and some which we have created ourselves.
- Systematically evaluating the performance of modern Object Detection algorithms on these datasets using custom pre-defined metrics.
- Comparing the robustness of the most prevalent Object Detection frameworks, one stage and two stage object detection, based on the previous results
- Extracting visual explanations of these results and providing custom metrics to quantitatively interpret these results.

- Offer possible interpretations and explanations on the models' performance according to the model and corruption type.

2.2 Computer Vision

Computer Vision is a field of Artificial Intelligence (AI) that has been getting increasing amounts of attention in the past few years. At its core, Computer Vision aims to replicate the complex human vision system and train computers to understand and interpret the visual world. Digital images, videos and other visual elements are processed and analyzed in order to extract meaningful representations that allow the system to derive logical conclusions regarding those elements and their context. Computer Vision tasks vary greatly in regards to their application as well as to the various techniques used to complete them. Examples of such tasks include Object Detection, Pose Estimation, Instance Segmentation, Scene Reconstruction etc. The most impressive applications of Computer Vision can be found in healthcare, where Machine Learning (ML) models have managed to detect cancers using MRI images [55], and produce valuable prognoses about patients' probabilities to develop certain diseases [28], as well as other industries like self-driving cars [11] etc.

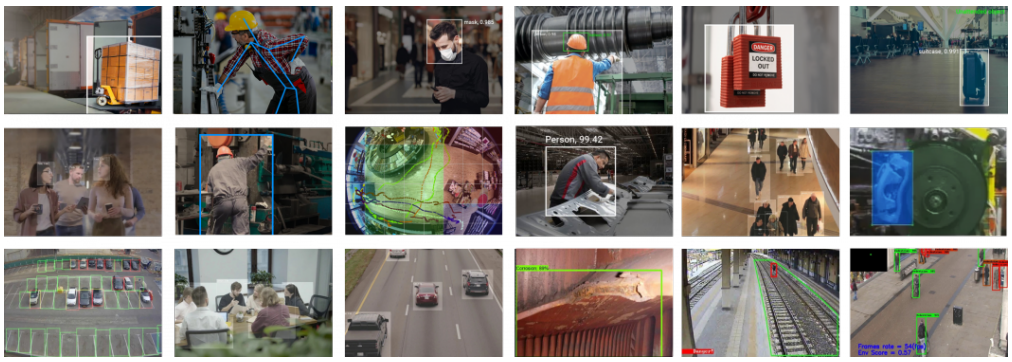


Figure 2.2.1: The various applications of Computer Vision

2.3 Object Detection

Object detection is perhaps the most prevalent task in the field of Computer Vision. It entails detecting instances of objects of a certain class within an image, encapsulating two different tasks: object localization and image classification. In object localization the algorithm outputs a set of (x,y) coordinates that define a bounding box that contains a certain object. Next, the algorithm must predict a class label for each object it detected, i.e. image classification. Although this task might sound simple, it is one of the most basic tasks in computer vision and therefore is still a very widely researched and competitive field, with new models that offer innovative ideas being introduced frequently.

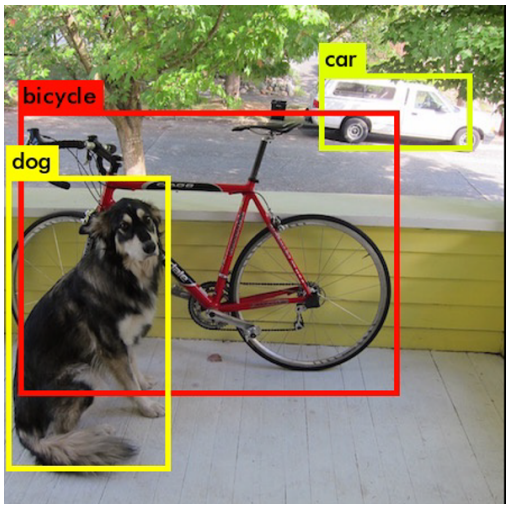


Figure 2.3.1: Object Detection using bounding boxes

As previously mentioned, Object detection finds countless real-world applications in a large variety of domains, such as bio-medicine, healthcare, agriculture, transportation, autonomous driving etc. This field has also been boosted significantly thanks to the great advances in imaging technology, with cameras getting smaller and cheaper, while providing better image quality, the dramatic increase in computer power, with Graphical Processing Units becoming stronger than ever, and the improved capacity of cloud platforms to host huge amounts of data. At the same time, object detection algorithms are becoming more advanced than ever, with new technologies constantly being implemented in order to push state-of-the-art performance even further. The state-of-the-art methods that tackle this task can be categorized in two main algorithm families: one-shot and two-shot algorithms, often called one-stage and two-stage algorithms, respectively. One-shot object detectors are usually faster, sometimes sacrificing inference accuracy, while two-shot detectors prioritize accuracy over speed. The most well known one-shot object detectors are the YOLO models [40], followed by SSD [25], EfficientDet [47] and RetinaNet [24], whereas the two-shot object detector category is dominated by the R-CNN [14] algorithm family, which includes Mask R-CNN, Fast and Faster R-CNN, Cascade R-CNN etc. Despite the differences in architecture, most object detectors follow a similar structure, so a standard model consists of three main parts: the backbone, or the feature extractor that extracts the feature map of the image, the neck, or the feature aggregator, and finally the head, which is the actual object detector, determining the bounding box and class label for each object in the image. These models are key in our analysis of Object Detection so they will be described more extensively in subsequent chapters.

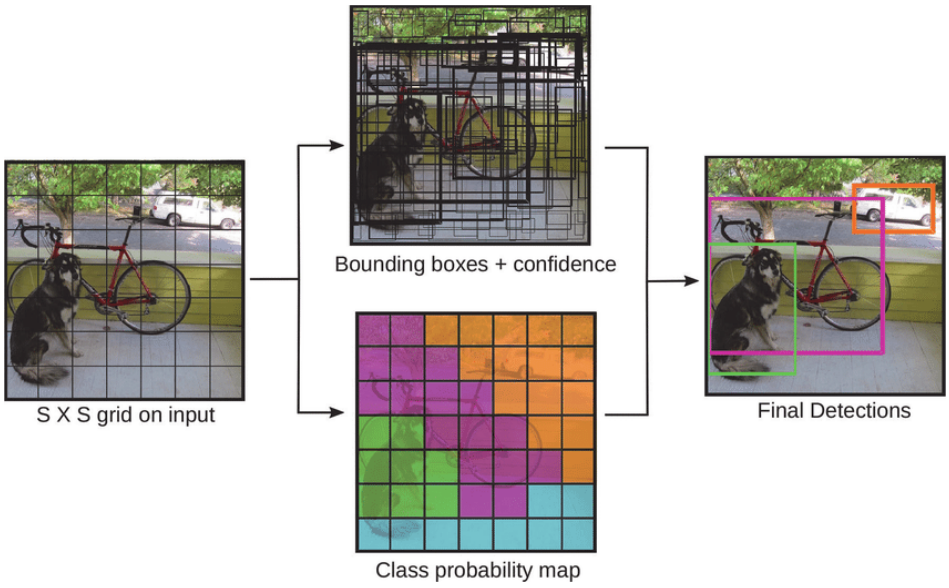


Figure 2.3.2: Example of two-stage Object Detection

An integral part of the development of Object Detection models are the datasets that are used to train and finetune them. These datasets vary in number and size according to the subtask, whether that is object detection in indoor spaces, outdoor spaces, object detection in autonomous driving etc. The most popular among these benchmarks is the Microsoft COCO (Common Objects in Context) dataset [7], a large-scale, public dataset that consists of 328,000 images of everyday scenes, both indoor and outdoor, and annotations for many different computer vision tasks such as object detection, segmentation, image captioning, and keypoint tracking. There are 80 object classes to be detected, ranging from people, animals and vehicles to common household objects like a hairbrush or a suitcase. This dataset is very prominent and most models are evaluated on its validation set since the labels for the test data are not publicly available. A similar, but much smaller, dataset that is also widely used in this context is the PASCAL Visual Object Classes Challenge (VOC) [8] dataset, which contains 20 everyday object classes found in 3000 labeled images. These datasets are ideal for the development of general-purpose models that aim to offer a general understanding of the space a person can find themselves in in their everyday life. On the other hand, there are datasets like KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) [10] that consist of hours of traffic scenes and that are targeted towards the development of robotic navigation systems, for which object detection is a crucial part. Considering the widespread use of object detectors in many real-life applications, the importance of large, reliably annotated datasets that help algorithm robustness becomes evident. Part of the evaluation of object detection models are also the pre-defined metrics that determine the models' performance compared to previous attempts at the same tasks. Since object detection consists of two sub-tasks, object localization and image classification, there are two seperate metrics that can be combined to help us evaluate the performance as a whole. In the task of object localization the detector has to define a bounding box that contains an object and the goal is for that bounding box to overlap with the ground-truth bounding box that has been previously annotated as much as possible. The main metric used to determine whether a bounding box prediction is accurate is the Intersection over Union (IoU) metric that can be defined as:

$$IoU = \frac{AreaofOverlap}{AreaofUnion} \tag{2.3.1}$$

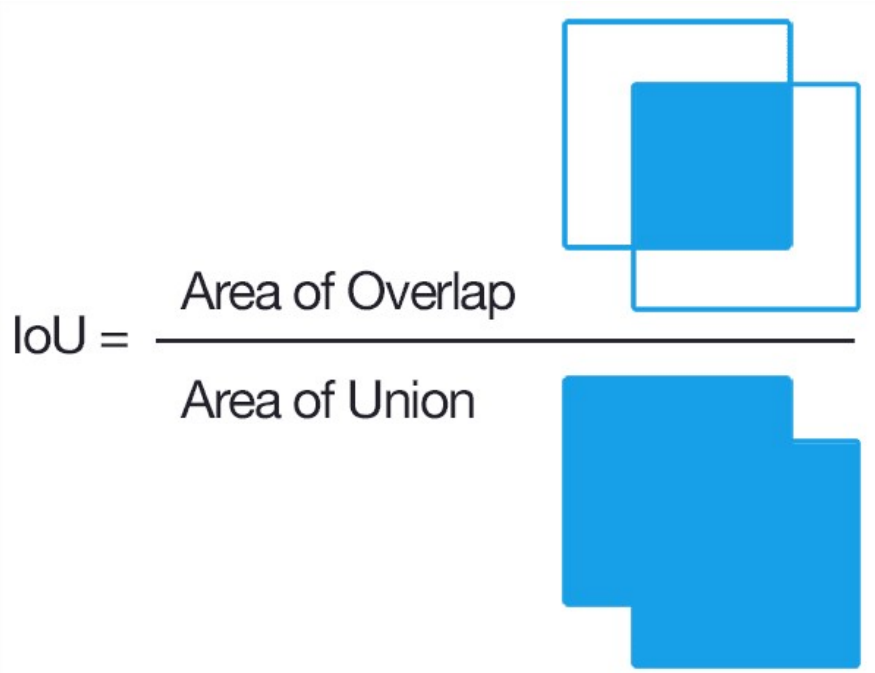


Figure 2.3.3: The IoU metric

So, a larger IoU suggests that there is a bigger overlap between the ground truth and the prediction. However, this metric is not enough for the evaluation of object detectors since the class label also needs to correspond to the ground truth. To that end, the mean Average Precision (mAP) is calculated for all object classes in a dataset. The general definition for Average Precision is the area under the precision-recall curve:

$$AP = \int_0^1 p(r)dr \tag{2.3.2}$$

Different datasets use different methods to smooth and sample the precision-recall curve, which leads to different definitions of the AP score. For the COCO dataset, which will be the focus of our experiments, AP is defined as the average of the APs calculated for a range of different IoU scores. So, an AP@[.50:.05:.95] (which is used for the COCO challenge and as an evaluation tool for most of the models we will use) corresponds to the AP calculated for an IoU threshold ranging from 0.5 to 0.95 with a 0.05 step.

After having described the necessary principles, datasets and evaluation metrics for the Object Detection task we will move forward by examining the most important models that have been developed for this task according to the one-stage vs two-stage detection paradigm.

2.4 The R-CNN Models

On the side of two-shot detectors there is the R-CNN family of algorithms, which stands for Region-Based Convolutional Neural Networks. As the name suggests, these models perform object detection in two stages: first a seperate algorithm generates a set of candidate regions of the input image that have a high probability of including an object, then these regions are input into a Convolutional Neural Network (CNN) and that CNN will output classification scores for each of these regions, deciding whether they include an object of a particular class based on an IoU overlap threshold. Let's take a look at the models that were developed as part of this family over the years.

- The original R-CNN model [14] was introduced in 2014, breaking the plateau that object detection models had reached at that point. It offered a 30% mAP increase on the PASCAL VOC dataset compared to the previous state of the art model, scoring a 53.3% mAP. The region proposals were

generated using Selective Search, an algorithm that groups together regions of an image based on their pixel intensities, and then input into different CNNs that extract feature vectors for each region, regardless of its size, by warping all pixels to a pre-determined size. Lastly, all regions are classified using pre-trained class-specific linear SVMs that classify a region proposal as one of the object classes, or as background.

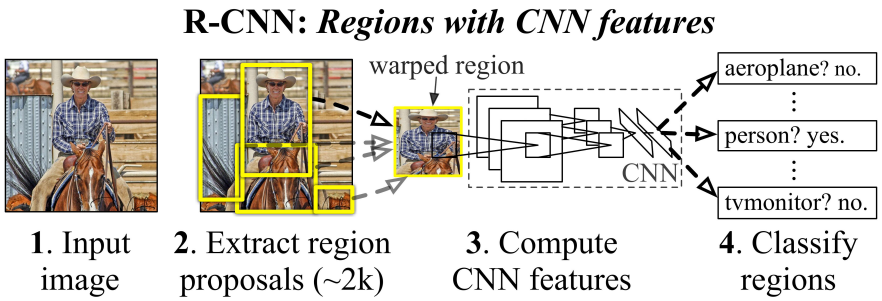


Figure 2.4.1: Original R-CNN Model Architecture

In this particular instance of the R-CNN model, 2000 bottom-up region proposals were generated for each image, and a 4096-dimensional feature vector was extracted for each region, while the extracted features from the CNNs needed to be cached in order to later train the SVMs. So, although this algorithm achieved state-of-the-art results, it is evident that the training and evaluation times become exorbitant for large datasets, combined with the enormous disk size needed to cache the extracted features. This is where the next version came into play.

- The Fast-RCNN [13] algorithm was developed in 2015 to combat the issues of the original R-CNN model, by using a single CNN model to extract features from all the regions. The network accepts an image and the corresponding region proposals as input and first the image feature map is extracted using convolutional and max pooling layers. Then, for each region proposal, a Region of Interest (RoI) pooling layer extracts a fixed-length feature vector that is then fed into a series of fully connected layers that generate probability estimates for K object classes. This network managed to increase R-CNN mAP on the PASCAL VOC, while decreasing training and inference times and eliminating the need for disk storage for feature caching.

- Faster-RCNN [41], introduced in 2016, is an extension of Fast-RCNN that aimed to further improve its speed and performance. The main contribution of this model is that it replaced Selective Search with a Region Proposal Network (RPN), which is a Fully Convolutional Network that predicts bounding boxes and objectness scores in various scales. This change was combined with the introduction of anchor boxes, which are reference boxes associated with different scale and aspect ratios All region proposals were generated relative to these anchor boxes, thus allowing the model to detect objects at different scales and creating a pyramid of anchor boxes. The overall network consists of the RPN and Fast-RCNN, which is responsible for detecting objects in the region proposals generated by the RPN, by classifying the extracted feature vectors. The model can be trained using three methods: 1)Alternating Training, in which the RPN is trained first and then the shared weights of the Fast-RCNN module are initiliazed, while the other weights are trained and tuned along with the RPN weights. 2)Approximate Joint Training, in which the two modules are trained as a single network and the weights are updated sequentially. 3)Non-Approximate Joint Training, in which a RoI Warping layer is used. This model achieved state of the art results not only on the PASCAL VOC but also on the MS-COCO dataset.

- Mask R-CNN [17] is not the final rendition of the R-CNN algorithm family, although it is the last to be mentioned/analyzed in this study. It was released in 2017 and included an extension for instance segmentation by returning a mask for each object detected along with the anchored bounding box and the class label. The same RPN as in Faster-RCNN is used for the first stage, while in the second stage a binary mask that encodes each object's spatial layout is calculated for each region proposed. The RoI pooling layer is replaced by RoIAlign, which aligns the extracted features with the input. Although

this model adds a small time overhead to Faster-RCNN, its extention of this popular architecture into different tasks such as keypoint detection, pose estimation etc. is worth noting.

2.5 The YOLO Models

The most prevalent one-shot object detection algorithms are the YOLO (You Only Look Once) algorithms, with the first model being introduced in 2015 [40] and the latest version, YOLOv8, being released in 2023 [54]. The principle that distinguishes these models is the lack of the region proposal stage that characterizes the two-stage object detectors, to which the faster inference speed of the one-shot detectors is attributed. Although the models have undergone many changes over the years and over different authors, the principles of one-shot object detection remain the same. Starting from the original YOLO model, the algorithm divides the input image into an SxS grid and if the center of an object falls into a grid cell then that grid cell is responsible for detecting that object. Each cell predicts B bounding boxes and their respective confidence scores, which reflect the model's confidence that a bounding box contains an object and how accurate that bounding box is.

The confidence score can be calculated as

$$Confidence = Pr(Object) * IOU_pred^truth \tag{2.5.1}$$

Another key technique that is used in the YOLO models is Non-Maximum Suppression (NMS), whose purpose is to select one of out of the many overlapping bounding boxes that are produced based on a pre-defined criterion, usually based on the IoU overlap between the predictions. The network architecture is based on the GoogLeNet model for image classification.

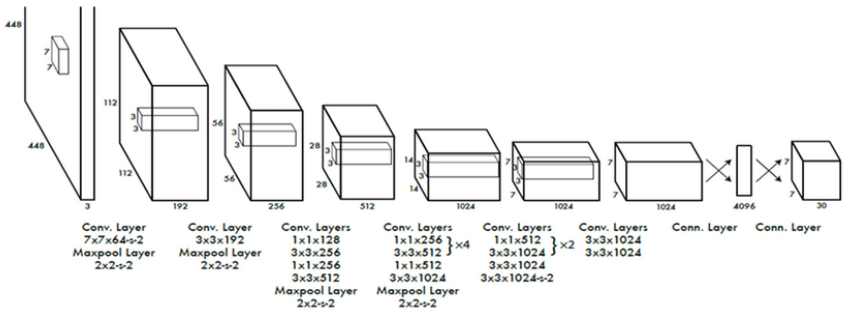


Figure 2.5.1: Original YOLO Model Architecture

All subsequent version of YOLO add significant improvements in different aspects of the architecture in order to boost performance, which we will summarize below:

- YOLOv2, or YOLO9000 [38] was introduced in 2016 and included a different backbone, Darknet-19, a variant of VGGNet but also introduced the use of anchor boxes. Anchor boxes are a set of predefined bounding boxes of different sizes that are used to capture the scale and aspect ratio of the specific classes to be predicted. Then, the predicted bounding boxes are not calculated directly but as displacements from the predetermined anchor boxes.
- In the third version, YOLOv3 [39], the model became larger and more accurate while maintaining superior speed compared to other detectors of its time. The main improvements employed are the addition of Feature Pyramid Networks [reference] to extract features from different scales and sizes from the same image, the improvement of the backbone to Darknet-53 as well as the addition of logistic classifiers to improve accuracy.
- YOLOv4 [2] was the first version of the YOLO models that was not developed by the original author, Joseph Redmon, who decided to step back from the projects due to concerns about the ethical usage of Computer Vision models. The new authors, A. Bochkovskiy et al., made many new contributions

to these models, first by introducing two new concepts: Bag of Freebies (BoF) to refer to methods that improve performance by modifying only the training strategy while not increasing the inference cost, so mainly data augmentation techniques, and Bag of Specials (BoS) to refer to post-processing methods that increase the inference cost but significantly improve the detection accuracy. Several data augmentation techniques were added to the BoF, including CutMix and two new methods: a) Mosaic, that mixes 4 training images to include 4 different contexts and b) Self-Adversarial Training (SAT), where the model performs an adversarial attack on itself by altering the input image and then detecting an object in that image. In addition to data augmentation, the BoF also included different types or regularization techniques such as Dropout, DropPath, Spatial dropout etc. and different types of normalization techniques, like the newly introduced Cross mini-Batch Normalization (CmBN) that collects statistics only within mini-batches in a single batch. The changes made to the BoS include the addition of Skip-Connections such as Cross stage partial connections (CSP) and the modification of other techniques, such as Spatial Attention Modules (SAM) that generates feature maps by utilizing their inter-spatial feature relationship.

- YOLOv5 [53] was released by the Ultralytics group in 2020 as a PyTorch implementation of YOLOv3, but no actual paper has still been produced to accompany this work. However, this is still one of the most popular versions of the YOLO models, scoring an 50.7% mAP score on the COCO validation dataset.

- The year 2022 saw two new releases of the YOLO algorithm, the first of which is YOLOv6, [22] published by yet another set of authors that introduced some fundamental changes. First, the architecture was switched from the Darknet architecture that was used as the backbone in all previous versions, to a dual approach that was named EfficientRep and included using a RepVGG backbone for the smaller network, as it is equipped with more feature representation power with a similar inference speed (but has prohibitively high computational costs in larger networks), and a revised CSP block named CSPStackRep for the large networks. Another main modification that impacted both the inference speed and performance is the transition from anchor boxes to anchor-free detection, specifically anchor point-based detection, where the distance from the anchor point to all four sides of the bounding box is predicted. With the addition of an SIoU bounding box regression loss that contains four cost functions (angle cost, distance cost, shape cost, IoU cost) and other finetuned elements, this model's smallest version achieves a score of 35.9 mAP on the COCO validation set, while the largest achieves a mAP of 57.2.

- Next, YOLOv7 [49], also published in 2022 by the same authors as YOLOv4, introduced an ensemble technique that merges different computational modules at the inference stage and surpassed all known object detectors of that time in real-time object detection. The BoF from YOLOv4 was upgraded to a Trainable Bag of Freebies that included a RepConv backbone without identity connection, named RepConvN, and a re-parameterization of the convolutional layers. The largest version of this network, YOLOv7-E6E, is comprised of 151.7 million parameters and achieves a score of 56.8% mAP on the COCO validation set, while the smallest, YOLOv7-tiny scores a 35.2%.

- Finally, YOLOv8 [54], the latest rendition of the YOLO lineage (at least at the time of writing) was released in 2023 by the Ultralytics group as a PyTorch implementation, again without a corresponding paper having been released yet, and the largest model, YOLOv8x achieves a 53.9% mAP score on the COCO validation set.

The performance of the different YOLO models on the COCO dataset has been incrementally increasing over the years, with the latest versions surpassing all other modern object detectors both in terms of speed and accuracy.

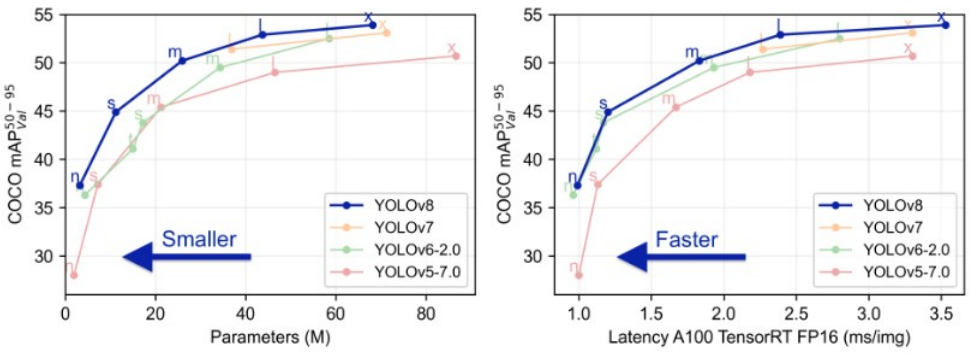


Figure 2.5.2: The evolution of the latest YOLO models [54]

2.6 Saliency Maps

Despite the many evaluation metrics that have been employed for this task, the way Object Detectors truly function and the reasons they make the decisions they do are still unclear to humans. And with the increasing role that these, and other Machine Learning models, play in modern society the need for transparency in the way they work is more urgent than ever. This is how Explainability emerged as a field of Artificial Intelligence that attempts to explain to humans how a model functions, from input to output, and how it comes to make a decision. If ML models are to be included in the decision making process of areas like Law or Medicine, it is essential that they be made explainable, to avoid assimilating their possible biases in our culture and also to increase faith in them and ensure they serve the good of the public. In the field of Object Detection the explainability methods tend to be visual, due to the nature of task, but there are also analytical methods that determine the contribution of a feature to the model decision. In this work, we will focus on visual explainability methods, and more specifically Saliency Maps. Saliency maps are heatmaps that represent how each pixel of an image affects the detectors' decision. Another interpretation of Saliency Maps is that they aim to depict on which parts of an image a person's eyes focus first. A highlighted region in a saliency map might imply that the actual object detected is located there, or that the region contains an important piece of context for an object that was detected. Some algorithms rely on static image features to localize the regions of interest in an image, while others that use video input consider objects that move as salient. However, in our study, the point of interest is not what elements a human will notice first in an image but an object detector. This study may unveil possible biases in our models like reliance on color, context and environment clues, or even possible biases in the datasets used to train these models.

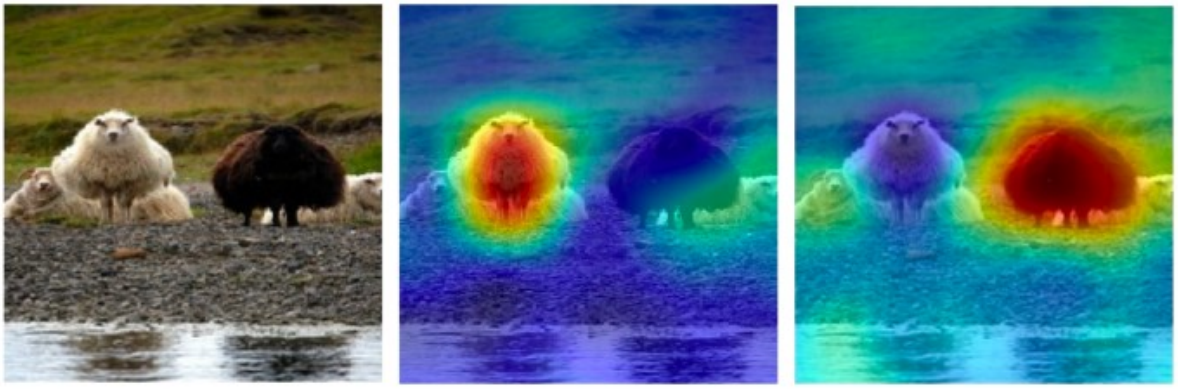


Figure 2.6.1: Examples of Saliency Maps [34]

The most popular approaches for creating saliency maps can be categorized in the following manner:

- **Gradient-based methods:** These methods utilize the gradient of a deep learning network to identify which parts of an image are most important for a particular decision. Backpropagation can be used to calculate the gradient of a network's output with respect to its input image, which can then be used to create a saliency map.
- **Perturbation-based methods:** These approaches involve making small changes to an input image and observing how the model's decisions change as a result. By analyzing these changes, it is possible to identify which parts of the image are most important for these decisions.
- **Activation-based methods:** The methods that belong in this category rely on the activation maps to complete this task by identifying which parts of the input image correspond to the highest activations in the network's activation map, as these regions are assumed to be the most important regions for the network's output.

Chapter 3

Related Work

3.1 Image Corruptions

The field of robustness in Object Detection, Image Classification and Instance Segmentation has been widely explored with many different methods that highlight the many aspects of these models that remain vague and unclear to humans. The first and most common group of works contains evaluating the performance of pre-trained model on perturbed or corrupted images in order to observe their deteriorating performance, locate gaps in their training according to the specific characteristics of each corruption, and fill in those gaps with data augmentation techniques, or by altering the network itself. In [18] the authors introduce a set of image corruptions benchmarks for testing the robustness of several image classifiers against common perturbations like the added effect of snow, rain, fog, different types of noises and blurs etc. They created 15 different perturbations, each of which has five levels of severity, resulting in 75 corruptions that they applied to ImageNet, a popular image classification dataset, to create two variations IMAGENET-C and IMAGENET-P. Next, they tested the performance of deep learning image classification models like AlexNet, ResNet, VGG etc. They also introduced a new set of metrics for evaluating this performance and documented deteriorating and unexpected results from all classifiers, with larger models such as ResNet being more robust, while also introducing a set of techniques that improves robustness against corruptions and perturbations. This work shares our goals and sets a very important benchmark for evaluating and improving classifier robustness, and could be generalized to fit the Object Detection task. A different study on deep learning image classifiers (ResNet, VGG, GoogLeNet) that tested their ability to generalize when an image is corrupted against a human's ability, showed that the human visual system is almost always superior when it comes to robustness, as it performed better on twelve different types of image corruptions. It was also noted that classifiers trained on distorted images outperformed humans on these exact distortion types, but were unable to generalize on other types, highlighting the challenges of robustness in the field. [12] Moving onto Object Detection, a work that investigates the vulnerability of deep learning models against image corruptions is [33], in which three perturbed benchmark datasets PASCAL-C, COCO-C and Cityscapes-C are introduced for Object Detection and specifically for autonomous driving, which is a prime example of a real-life application where Object Detectors need to be able to adjust to various weather conditions without necessarily having been trained on such images. Following the work of Hendrycks et al., they used 15 different types of corruptions, including added blur, noise, frost, fog etc., with 5 degrees of severity, to test popular object detectors such as Faster R-CNN, Mask R-CNN, Cascade R-CNN etc. They also introduced their own evaluation metrics and reported that a stronger backbone causes performance improvement on corrupted data, whereas a more powerful head does not. Lastly, they propose a new method of stylizing images during training which leads to improved robustness against corruptions during evaluation time. Our work is an extension of this study to include state of the art one-stage object detectors like the YOLO line of models, since we have utilized their images corruptions to apply to the COCO validation set, while also introducing a few new corruptions and extending the analysis by leveraging the use of saliency maps. There have also been attempts to evaluate the robustness of image segmentation models [1], [21], in which the same corruptions by Hendrycks et al. are applied on images to test instance and semantic segmentation models respectively, benchmarking the performance of different backbones and suggesting possible solutions to increase robustness in these models.

3.2 Adversarial Attacks

Another emerging field on Machine Learning that has been gaining significant popularity and has been applied to robustness study is Adversarial Machine Learning. The term Adversarial Attack refers to a technique that aims to "fool" a model using deceptive data, either during training or during inference. These techniques are based either on modifying the input data in a way that is imperceptible by a human observer or by directing altering the model's parameters, thus causing it to malfunction. These are classified as black-box and white-box attacks respectively, however we will be focusing on black-box attacks since they are much more common and do not require specific knowledge of a model's functionality. A black-box adversarial attack against an object detection model can include adding imperceptible noise to an image that will cause the detector to malfunction and identify an object as something completely different, by feeding an online model with maliciously tampered data, or even by altering a single pixel of the input image [46]. However, this set of methods can also be used to increase the robustness of a model when used as a data augmentation technique. In [5] the authors integrate adversarial examples into a data augmentation technique that is

used during the fine-tuning stage of a detector and apply it separately to the image classification and object localization branches. More specifically, they make use of the corruption types introduced in [18] and feed different EfficientDet models both with clean and corrupted images and observe not only that the model has slightly increased accuracy but also that it is more robust to image distortions. On the other hand, robustness against adversarial attacks in image classifiers and object detectors has also been studied extensively [26], [32], [31], [51], [16], [36], [52], [6], however this field of study falls outside the scope of this work.

3.3 Distribution Shifts

In the field of Image Classification, and therefore Object Detection, the concept of distribution shifts as tools for testing and increasing model robustness has been gaining increasing popularity in recent years. The term distribution shift in general was created to describe the phenomenon where the underlying distribution of the data the model was trained on differs greatly from the data it is tested or deployed on. In the context of image classification and object detection, this could refer to testing a model on images that contain unseen object types, changed features, entirely different scenes, different lighting conditions or camera settings, a different camera perspective etc. Ideally, any ML model and therefore any classifier or detector would be able to adapt to distribution shifts, whether they are natural or artificial, however research has shown that state-of-the-art object detectors and image classifiers are vulnerable even to small changes in the input distribution and perform poorly on out-of-distribution data [37], [48], [19], [3], [30]. Particularly in the Computer Vision domain, where models are typically deployed in real-world scenarios with the input data coming from a variety of different sources and distributions, robustness against distribution shifts is crucial for increasing model reliability. Various works have been developed in the literature for studying the robustness of image classifiers and object detectors against distribution shifts using a wide variety of methods. In [30] the authors explore the problem of adapting object detection models to new domains with differing training and testing data distributions and propose a method for unsupervised domain adaptation that aligns the distributions of the target and source domains, which shows promise in improving the models' performance under distribution shifts. Another application of this is found in [23], where the authors propose a training schema in which different noisy labels are generated after each gradient update during model training, so the model does not overfit on a specific type of noise. Their approach trains the model to adapt to the distribution of the noisy data by learning to correct the noise, making it more robust against added noise in the data distribution during testing. A different approach is followed by the authors in [27], where they examine the performance of different image classifiers under distribution shifts that stem from uncurated images sourced from the web, unlike in other works where the data used to simulate the distribution shift are carefully selected. They observe the performance drop of these models on the collected dataset and explain the inability of simple accuracy metrics to capture the entire essence of this deterioration, so a knowledge-driven evaluation schema that captures the semantic relations between the misclassified samples is proposed. This type of analysis, that goes past strict accuracy-based evaluation and attempts to uncover deeper biases and relations in image classifiers and object detectors is what we hope to achieve in this work as well.

3.4 Saliency Maps

On the topic of saliency maps, many previous works rely on this, and other, methods to gain a better insight on the way object detectors operate, unveiling possible biases, such as people's skin color leading to reduced performance when it comes to dark-skinned individuals [50]. However, saliency maps can be directly used to increase model robustness. In [29] and in [20], saliency maps and other explainability methods are used to generate adversarial examples, resulting in improved performance as the adversarial attack got stronger and therefore increasing model robustness. Another functionality of saliency maps in robustness study can be found in [4], where it was shown that the saliency map produced based on a robustness-enchanced model was much more robust against perturbations compared to the saliency map produced based on a regular model. This approach has also been widely used to study the explainability of image classifiers in many works [44], [43], [56]. Particularly in [9] the authors attempt to produce a mask that represents the most important feature of an image according to image classifiers, by progressively applying certrain perturbations such as noise or blur, and observing how the classifier's decision changes. Our approach to robustness based on saliency maps is similar, in that it aims to observe the changes that occur as the model is fed progressively

more corrupted images, with different kinds of corruptions and attempting to quantify these results.

Chapter 4

Object Detection Experiments

4.1 The MS COCO Dataset

Many datasets have been introduced for this task over the years, including the PASCAL Visual Object Classes (VOC) Challenge, the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) mainly used in mobile robotics and autonomous driving etc. However, the most popular benchmark on this task is the Microsoft COCO (Common Objects in Context) dataset [7], a large-scale, public dataset that consists of 328,000 images of everyday scenes, and annotations for many different computer vision tasks such as object detection, segmentation, image captioning, and keypoint tracking. We will only be focusing on the object detection segment of the dataset, whose captions include 80 class labels of everyday objects, varying from people and vehicles to animals and household objects. This dataset will play an integral role in our experimentations, so we provide an analytical overview of the dataset and its classes.

person	elephant	wine glass	dining table
bicycle	bear	cup	toilet
car	zebra	fork	tv
motorcycle	giraffe	knife	laptop
airplane	backpack	spoon	mouse
bus	umbrella	bowl	remote
train	handbag	banana	keyboard
truck	tie	apple	cell phone
boat	suitcase	sandwich	microwave
traffic light	frisbee	orange	oven
fire hydrant	skis	broccoli	toaster
stop sign	snowboard	carrot	sink
parking meter	sports ball	hot dog	refrigerator
bench	kite	pizza	book
bird	baseball bat	donut	clock
cat	baseball glove	cake	vase
dog	skateboard	chair	scissors
horse	surfboard	couch	teddy bear
sheep	tennis racket	potted plant	hair drier
cow	bottle	bed	toothbrush

Table 4.1: Available classes in the COCO dataset

Class	**Instances in the Train Set**	**Instances in the Val Set**
Person	262.465	11.004
Car	43.867	1.932
Chair	38.491	1.791
Book	24.715	1.161
Bottle	24.342	1.025
Cup	20.650	899
Dining Table	15.714	697
Bowl	14.358	637
Traffic Light	12.884	626
Handbag	12.354	540
Umbrella	11.431	413
Bird	10.806	440
Boat	10.759	430
Truck	9.973	415
Bench	9.838	413

Table 4.2: Top 10 most frequently appearing classes in the COCO dataset

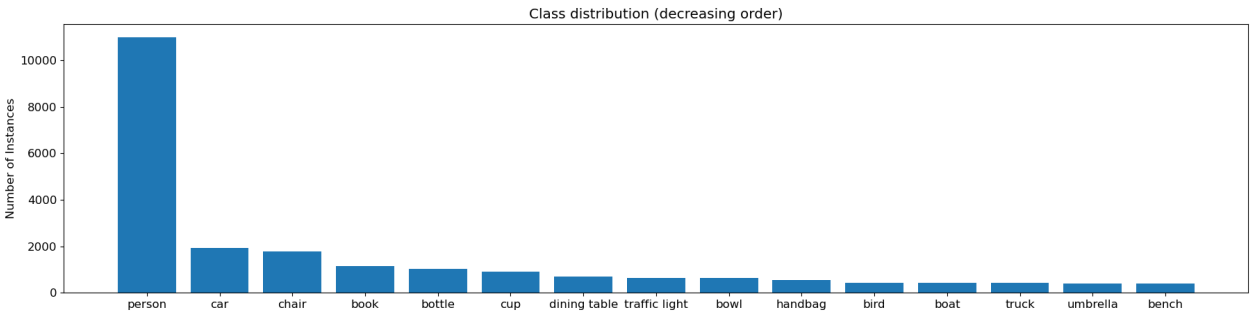


Figure 4.1.1: Number of Instances per Class in COCO training set

Most Object Detection models are trained, validated or tested on a subset of the COCO dataset, and new works that are being released publish their performance on it in order to show the importance of their work. The current best performing models on this dataset are as follows:

Model	AP Score	Release Year
YOLOv6-L6	57.2	2023
YOLOv7-E6E	56.8	2022
YOLOv7-D6	56.6	2022
YOLOv7-E6	56	2022
YOLOv7-W6	54.9	2022
YOLOv7-X	53.1	2022
YOLOv7 Table	51.4	2022
YOLOv5-X	50.4	2021
Mask R-CNN	45.2	2017

Table 4.3: Best performing recent models of the COCO dataset

It is obvious that the field of Object Detection over the past years has been dominated by YOLO and its latest release, which is why we have selected this model family to be the main source of experimentation, with Mask R-CNN following, although not closely given its release year, mostly to provide with a baseline comparison between one shot and two shot object detectors.

4.2 Experiment Pipeline

In order to gauge how robust modern Object Detectors are to changes in image quality, whether that comes from extreme weather conditions, applied noise due to camera malfunctions, blur from abrupt camera movements or other alterations, we need to establish a baseline performance of these models and then methodically analyze the way it deteriorates according to each perturbation. To this end, we will utilize the MS COCO dataset, since it is arguably the most popular dataset on this task and most researchers use it to validate their models' performance compared to others, while also containing a variety of indoor and outdoor images and various object classes. We will begin our experimentation by establishing the already known performance of our chosen detectors on the original COCO validation set, since the labels of the test set are not publicly available. Next, we will apply various types of perturbations on these images, with varying degrees of severity, and document the decline in performance of each detector according to the nature of each perturbation. After these results are documented and analyzed we will select the most interesting/conflicting aspects and move on to the next step of our experimentation which will be the visualization. We will utilize Saliency Maps to gauge the way the regions of interest change according to the perturbations as well as the shift in the importance of contextual clues in the image.

4.2.1 Model Selection

In our experiments we will also be comparing the performance of both one-stage and two-stage detectors, since their architecture is fundamentally different, so it is expected that they will behave differently under the same corruptions. More specifically, we will select Mask R-CNN as the two-stage detector since it is one of latest algorithms of the R-CNN family and the YOLO models as the one stage detectors. We began our experimentation with YOLOv5 as it was the latest YOLO model at the time of writing, however, since then there have been many new releases of YOLO, so in order to provide a more comprehensive study we included every new version of YOLO as it came, so our model selection for the one-stage detectors consists of YOLOv5, YOLOv6, YOLOv7 and YOLOv8. Another point to note is that we will be testing two different versions of each of these models: the smallest available version, usually called the nano, and the largest available. The smaller models have fewer parameters, require less space and they are faster but less accurate, while the larger ones fall on the opposite side, with more parameters and higher inference times but increased accuracy. We are formatting our experiments this way to analyze how important model size and complexity is for robustness, as well as to establish a trade-off between inference speed and accuracy/robustness. Each version has a different use, e.g. nano models could be useful for mobile applications where speed is crucial, but all of them need to have established robustness. Lastly, the reason why we focus more on the one-stage detectors is because two-stage detectors, and mainly the R-CNN family, have been already studied extensively for their robustness, whereas there hasn't been a comprehensive analysis of robustness for the different YOLO models, to our knowledge. Also, the Object Detection stage has been dominated by YOLO models in recent years, which is why a benchmark of their robustness on a large variety of perturbations is important. However, we still test Mask R-CNN to provide a comparison between the different types of detectors, and because the R-CNN family had been dominating the Object Detection stage until the rise of the YOLO models. We provide an overview of the models we will be using for our experiments, including their inference speed, which is relevant for real-time object detection, or object detection on video.

Model	$mAP^{val}_{0.5:0.95}$	# of Params (M)
YOLOv5n	28.0	1.9
YOLOv5x	50.7	86.7
YOLOv6n	37.5	4.7
YOLOv6l	52.8	59.6
YOLOv7	51.4	37
YOLOv7E6	53.1	71.3
YOLOv8n	37.3	3.2
YOLOv8x	53.9	68.2
Mask RCNN	45.2	63.8

Table 4.4: Overview of Selected Models

The detector scoring the best performance on the COCO validation set among our selection is also the latest, YOLOv8x, which does not include the largest number of parameters. Note that, obviously the larger models will outperform the smaller ones in terms of accuracy, however we want to evaluate the models' robustness, meaning the rate at which they deteriorate under certain corruptions and not the absolute best performace accuracy-wise.

4.3 Image Corruptions

To perform our experiments we will use the corruptions provided by [18] and apply them to all 5000 images of the COCO validation set. These corruptions include:

- **Noise Corruptions**: Gaussian noise, Shot noise, Impulse noise
- **Blur Corruptions** : Defocus Blur, Glass Blur, Motion Blur, Zoom Blur
- **Weather Corruptions**: Snow, Frost, Fog, Brightness

- **Digital Corruptions**: Contrast, Elastic Transform, Pixelation, Jpeg Compression

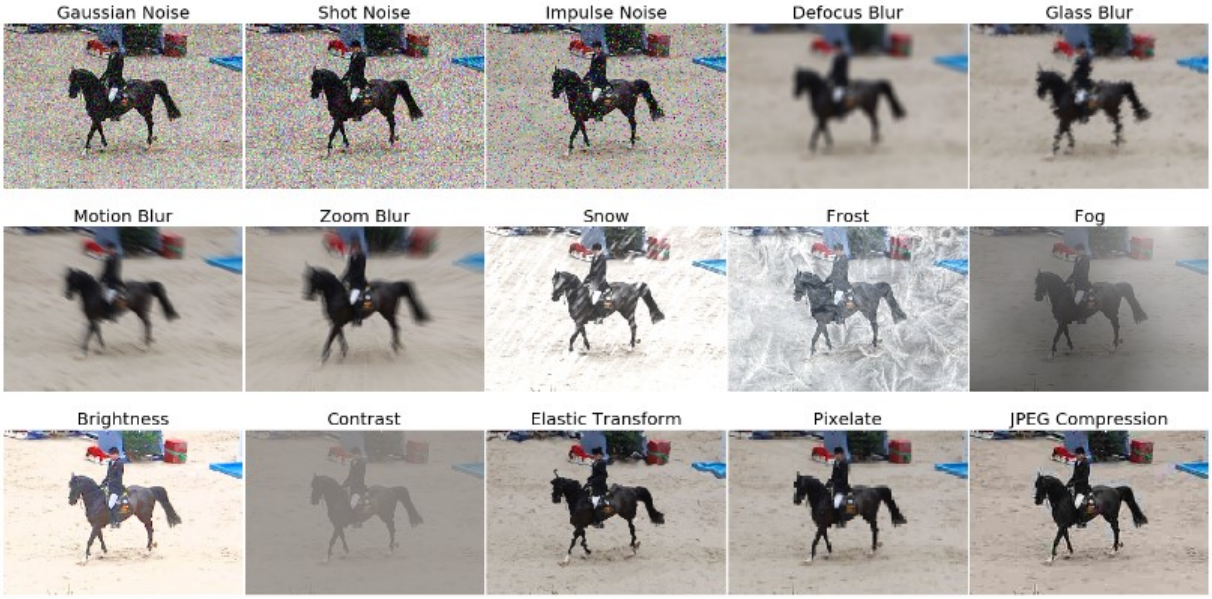


Figure 4.3.1: Original image corruptions introduced in [18]

In order to enrich the variability of the corruptions we introduce three new corruptions:

- **Rain**: Introduces an effect of rain to the image, ranging from a slight drizzle to torrential rain. We produce this corruption by using the imgaug library, defining five different deterministic rain augmentors and modifying their parameters so each of them matches a specific degree of rain severity, and then apply these augmentors to the COCO validation set. This is a basic weather corruption that needs to be part of our experiments, both from a theoretical standpoint but most importantly from a practical standpoint, as rain is very common and outdoor navigation systems (vision aids or autonomous vehicles) need to be able to achieve good performance while the image is affected by rain.

Figure 4.3.2: 5 levels of severity for our **Rain Corruption**

- **Dark**: This corruption progressively darkens the image to simulate a nighttime effect, by slowly lowering pixel values. Its aim is to establish how important color context is to an object detection model, since by slowly darkening the image the color of its environment disappears, while the edges of objects become less apparent and harder to distinguish.

Figure 4.3.3: 5 levels of severity for our **Darken Corruption**

- **Mask**: This corruption entails a random masking of parts of the image. For each severity level we define the total number number of pixels to be masked as N and a cluster factor C, then we select a number random pixels equal to N and mask a square patch of size $[C,C]$ of the original image. This algorithm is based on [15], where the authors employ the technique of masking to create a self-supervised audio transformer, however masking appears in many works and across many different domains. We define five different random mask generators and apply them to the COCO validation set. The purpose of this corruption is to determine the importance of contextual clues in object detectors, and whether they can produce the same results when most of the image environment is missing. This technique is also a popular pretraining technique for self-supervised networks.

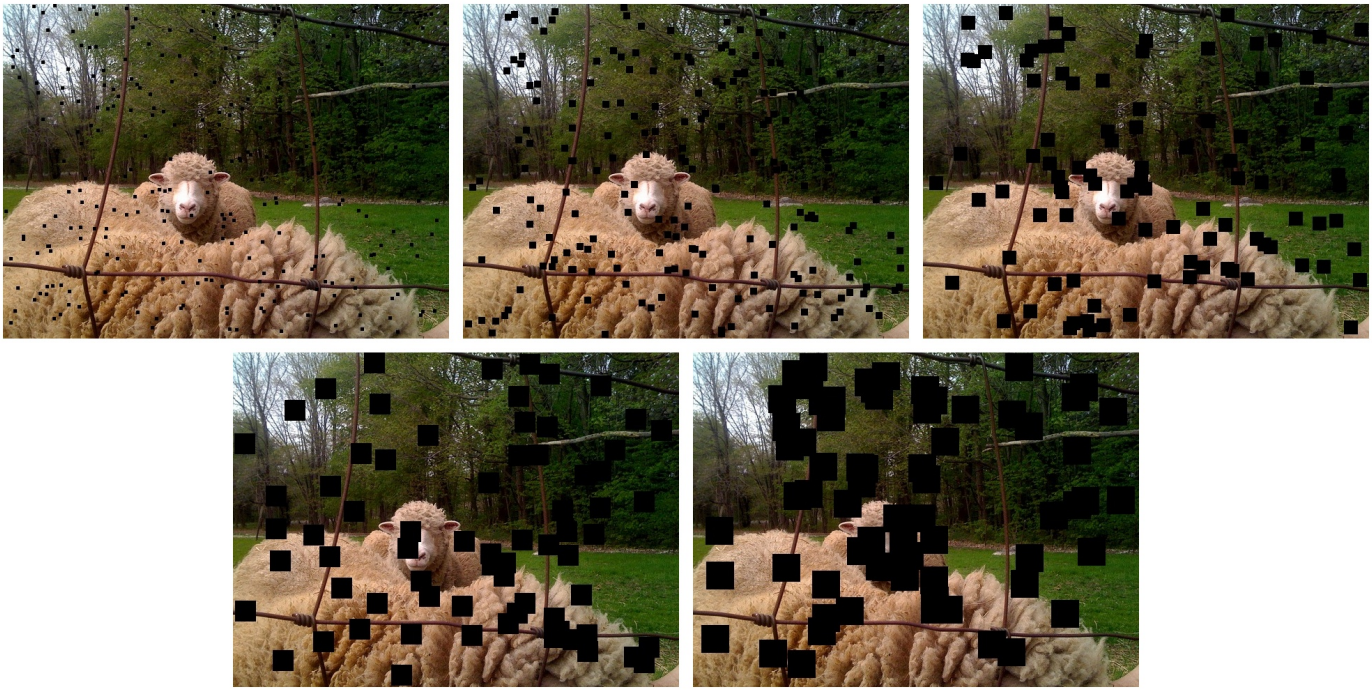


Figure 4.3.4: 5 levels of severity for our **Mask Corruption**

4.4 Results

We present a table containing results for all small detectors and all the corruptions for severity level 1. The tables for all severity levels, both for large and small detectors, can be found in the Appendix. We also present a graphical representation of our results for all corruptions and all severity levels for YOLOv5n, while the graphs for all other detectors can also be found in the appendix.

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.252	0.267	0.399	0.246	0.226
Frost	0.297	0.308	0.438	0.292	0.262
Fog	0.32	0.332	0.453	0.315	0.279
Brightness	0.357	0.362	0.48	0.24	0.345
Darken	0.349	0.356	0.468	0.347	0.315
Rain	0.339	0.345	0.468	0.336	0.325
Gauss	0.27	0.302	0.413	0.259	0.264
Impulse	0.224	0.28	0.366	0.221	0.194
Shot	0.27	0.306	0.412	0.262	0.263
Defocus	0.305	0.305	0.415	0.302	0.254
Zoom	0.131	0.131	0.208	0.127	0.108
Motion	0.289	0.306	0.409	0.289	0.269
Jpeg	0.282	0.313	0.347	0.277	0.263
Contrast	0.317	0.329	0.454	0.312	0.279
Pixelate	0.266	0.342	0.376	0.307	0.262
Elastic	0.298	0.318	0.418	0.308	0.278
Mask	0.274	0.251	0.404	0.257	0.255

Table 4.5: mAP Scores for **Small Detectors - Severity Level 1**

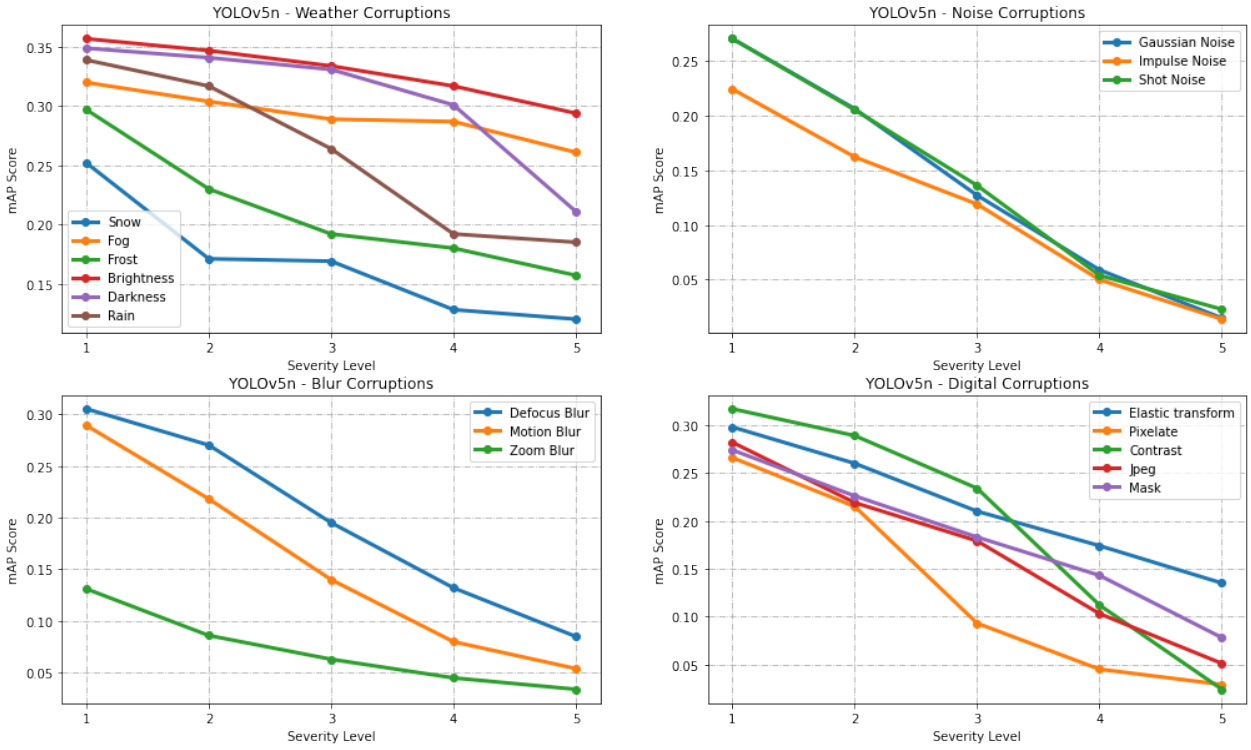


Figure 4.4.1: YOLOv5n performance on Corrupted COCO dataset

Lastly, we also define a metric that we name *GmAP* to get a more global evaluation of the robustness of each detector. Specifically, for each corruption we calculate the difference of the mAP score of each severity level with the following severity level, and finally get the average of these differences:

$$GmAP = \frac{\sum_{i=0}^{3}(mAP_{sev=i+1} - mAP_{sev=i})}{4} \tag{4.4.1}$$

A higher absolute GmAP value means that the differences in detector performance across the increasing severity levels was deteriorating faster compared to a detector with a lower absolute GmAP value, therefore an overall less robust performance of the detector on that specific corruption.

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.044	0.044	0.039	0.045	0.045
Frost	0.047	0.046	0.033	0.047	0.047
Fog	0.02	0.018	0.015	0.015	0.24
Brightness	0.019	0.017	0.021	0.021	0.035
Darken	0.046	0.042	0.04	0.049	0.067
Rain	0.051	0.046	0.028	0.056	0.051
Gauss	0.085	0.09	0.11	0.083	0.079
Impulse	0.07	0.083	0.096	0.07	0.055
Shot	0.082	0.086	0.108	0.082	0.076
Defocus	0.073	0.073	0.084	0.075	0.064
Zoom	0.032	0.032	0.047	0.031	0.028
Motion	0.078	0.082	0.104	0.079	0.073
Jpeg	0.077	0.071	0.087	0.072	0.071
Contrast	0.098	0.092	0.084	0.094	0.09
Pixelate	0.079	0.089	0.113	0.088	0.078
Elastic	0.054	0.057	0.072	0.054	0.052
Mask	0.065	0.064	0.06	0.064	0.057

Table 4.6: Absolute GmAP scores for all Small detectors and Corruptions

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.044	0.037	0.046	0.036
Frost	0.037	0.034	0.036	0.033
Fog	0.015	0.014	0.015	0.016
Brightness	0.021	0.016	0.019	0.019
Darken	0.05	0.042	0.048	0.042
Rain	0.03	0.024	0.03	0.029
Gauss	0.104	0.093	0.095	0.106
Impulse	0.095	0.083	0.082	0.093
Shot	0.099	0.086	0.09	0.1
Defocus	0.087	0.083	0.087	0.091
Zoom	0.049	0.051	0.049	0.052
Motion	0.104	0.098	0.1048	0.1
Jpeg	0.099	0.088	0.087	0.1
Contrast	0.108	0.09	0.095	0.088
Pixelate	0.121	0.113	0.12	0.118
Elastic	0.076	0.074	0.08	0.074
Mask	0.075	0.079	0.075	0.06

Table 4.7: Absolute GmAP scores for all Large Detectors and Corruptions

There are a lot of interesting observations to be made about these results, both for each detector separately but also collectively. First, it is clear and expected that all detectors' performance deteriorates as the corruption severity level increases. Secondly, when comparing the robustness of one-stage and two-stage algorithms, what is interesting to note is that the Mask R-CNN model is more robust than all versions of the small YOLO models on 9/15 total corruptions, despite being an older model. Particularly, we can see that Mask R-CNN scores the lowest GmAP values amongst all detectors on all the Blur corruptions, Impulse Noise, and the Jpeg, Pixelate, Elastic and Mask corruptions. This result could indicate that two-stage detectors

are more robust to certain kinds of corruptions than one-stage detectors, however in order to solidify such a conclusion we would need to experiment with a two-stage detector that matches the size of the large YOLO models available today. So far, we can only state that Mask R-CNN seems more robust against corruptions that distort the image without affecting its colors, particularly against blurring of the input image, and less robust against corruptions that change the color distribution (like the Brightness and Darken corruptions) and corruptions that directly insert new edges and shapes to the image (like most Weather corruptions). Now comparing between the small YOLO models, we observe that YOLOv7 is the most robust model against most Weather corruptions, but the least robust against the Noise and Blur corruptions. The YOLOv5n and YOLOv8n models perform similarly in terms of robustness, which is to be expected since they were developed by the same group, scoring better GmAP values than YOLOv7 against Blur corruptions and the Pixelate, Jpeg and Elastic corruptions that have the similar effect of pixelation on an image. On the other hand YOLOv6 seems to be among the best performers against Digital corruptions

When it comes to the large detectors, there seems to be no significant deviation in their GmAP scores, with all models achieving similar scores and outperforming each other by a slight margin in individual corruptions that follow no discernible pattern. However, what is interesting to note is that, while the large YOLO models always significantly outperform their smaller equivalents in terms of mAP score, which is expected as the large models often include an exponentially larger number of parameters, the same conclusion does not hold in terms of GmAP score. Specifically, YOLOv5n and YOLOv8n almost exclusively score better GmAP results compared to YOLOv5x and YOLOv8x respectively, and for YOLOv7, despite the smaller model still scoring better GmAP scores in most cases, the larger model scores better in more cases than the previous models. YOLOv6l on the other hand outperforms YOLOv6n when it comes to Weather corruptions, but falls short when it comes to all other categories, which evens out the robustness score between them. This observation is interesting in the sense that, the large detectors still produce better results no matter the corruption or severity and the expected result is that they would also produce better GmAP values, however that is not the case. Therefore, we can safely say that a larger parameter number does not ensure robustness in the detector's results. The consensus still stands that the larger models offer better accuracy against all corruptions and are therefore more robust, however when it comes to improving absolute robustness of a detector, a larger model does not equal a more robust model. This result leads us to the conclusion that in order to improve robustness in object detectors we cannot lean on simply larger models, which are accompanied by a significantly slower inference speed, but focus on how other elements can help improve robustness, such as pretraining and augmentation techniques.

Finally, we produce the mean GmAP score for all detectors across all corruptions, in order to arrive at a general conclusion about their overall robustness. We define the mean GmAP score as:

$$mGmAP = \frac{\sum_{i=corruption} GmAP_i}{\#\ Corruptions} \tag{4.4.2}$$

Once again, a smaller absolute mGmAP score points to a more robust detector. The final mGmAP results for all detectors are presented in the table below.

Detector	**mGmAP**
YOLOv5n	0.0601
YOLOv6n	0.0607
YOLOv7	0.067
YOLOv8n	0.06
Mask RCNN	**0.0583**
YOLOv5x	0.0713
YOLOv6l	0.065
YOLOv7E6	0.0681
YOLOv8x	0.06805

Table 4.8: Absolute mGmAP scores for Detectors

These results summarize all our previous observations. More specifically, Mask RCNN is the most absolutely

robust detector (meaning compared to itself), which leads us to assume that the trade-off between accuracy and inference speed between one-stage and two-stage detectors includes robustness as well, with two-stage detectors being overall more robust against different types of image corruptions. When it comes to one-stage detectors, the smaller models are more absolutely robust compared to their larger equivalents, with YOLOv8 being the more robust detector, both among the small and large models. Unfortunately, since we have no knowledge about this detector's architecture, pretraining or data augmentation techniques we cannot explain this superiority in performance.

Another interesting interpretation would be to determine which corruption from each category causes the worst results among all the classifiers, and which corruption is the worst overall. After having defined the metrics that summarize detector performance we will also propose a related metric that summarizes the effect of the different corruptions over all detectors, or over a specific model. Given a specific corruption C we denote the proposed metric as $CmAP$ and define it as:

$$CmAP = \frac{\sum_{i=0}^{N} GmAP_i}{N} \tag{4.4.3}$$

where $GmAP_i$ is the GmAP performance of detector i on this corruption, for N total number of detectors. Once again, the larger the absolute CmAP value, the worse effect this specific corruption has had on the detectors it was calculated on. We will calculate this metric for all corruptions, first on all detectors in order to determine which corruption caused the worst overall results and then repeat this process for each model (both the smaller and larger version) in order to document which corruption affects each classifier more. We present our results below, starting with the CmAP values for each corruption over all the detectors:

Corruption	**CmAP**
Snow	0.042
Frost	0.04
Fog	0.0173
Brightness	0.0208
Darken	0.0473
Rain	0.0383
Gauss	0.0938
Impulse	0.0807
Shot	0.0898
Defocus	0.796
Zoom	0.0412
Motion	0.09
Jpeg	0.084
Contrast	0.093
Pixelate	0.1
Elastic	0.0658
Mask	0.0665

Table 4.9: Absolute CmAP scores over all Detectors for all Corruptions

After observing these results we distinguish the Pixelate corruption as the one having the worst overall effect on our set of models, with the Gaussian Noise, Contrast and Shot Noise corruptions following shortly behind. In general, pixelation can lead to a loss of a lot of the fine details in an image and can make its edges and contours appear jagged, which could justify the large decrease in detector performance.

Overall we can see that the Noise corruptions achieved some of the highest CmAP scores, meaning they caused the worst decline in overall detector performance. This result is interesting since adding noise to the input image changes its underlying distribution, for example with Gaussian noise the resulting distribution of pixel values will likely be more spread out and have a wider range of values than the original distribution or with Impulse noise that randomly replaces some pixels in an image with either the minimum or maximum value

the change in the image distribution is drastic because this particular type of noise creates sudden spikes or dips in the pixel values. But the most important pattern that seems to be emerging from these results is that corruptions that affect the edges and texture of the image making it appear more homogeneous, which can make it harder for object detection algorithms to distinguish between different objects and backgrounds, cause an overall more significant decline in accuracy and robustness. This hypothesis also justifies the observation that smoother corruptions such as Brightness, Fog, Darkness etc. do not downgrade model performance as much. Given the case of the Mask corruption, which at the final severity level masks an important percentage of the image and still does not score at the top worst corruptions, we could postulate that discrete object edges are more important than context clues when it comes to object detection.

In the case of the Contrast corruption, has a highly adverse effect on detectors, we can attribute this phenomenon to the fact that contrast adjustments can make the features of objects in the image appear more pronounced or subdued, which can cause the detector to detect false positives due to exaggerated features or miss objects altogether due to subdued features, leading to false negatives. Moreover, contrast adjustments can alter the lighting and shadowing in an image. This can make it more difficult for the object detector to accurately distinguish objects from the background, especially if the contrast adjustment results in areas of the image being overexposed or underexposed. Finally, contrast adjustments can also introduce noise or artifacts into an image, especially in areas of high contrast. This can cause the object detector to detect false positives or miss objects that have been obscured by the noise or artifacts.

We will be repeating this analysis for each model specifically, in order to arrive at a conclusion about which corruption affects which detector more and whether some corruptions affect all detectors equally. To that end, we will be calculating the CmAP scores of each corruption for each model separately, but jointly among small and large versions, and comparing them.

Detector	**YOLOv5**	**YOLOv6**	**YOLOv7**	**YOLOv8**	**Mask RCNN**
Snow	0.044	0.0405	0.0425	0.0405	0.0225
Frost	0.042	0.04	0.0345	0.04	0.0235
Fog	0.0175	0.016	0.015	0.0175	0.012
Brightness	0.02	0.0165	0.02	0.02	0.0175
Darken	0.048	0.042	0.044	0.0455	0.0335
Rain	0.0405	0.035	0.029	0.0425	0.0255
Gauss	0.0945	0.0915	0.1025	0.0945	0.0395
Impulse	0.0825	0.083	0.089	0.0815	0.0275
Shot	0.0905	0.086	0.099	0.091	0.038
Defocus	0.08	0.078	0.0855	0.083	0.032
Zoom	0.0405	0.0415	0.048	0.0415	0.014
Motion	0.091	0.09	0.098	0.0895	0.0365
Jpeg	0.088	0.0795	0.0925	0.086	0.0355
Contrast	**0.103**	0.091	0.0895	0.091	**0.045**
Pixelate	0.1	**0.101**	**0.1165**	**0.103**	0.039
Elastic	0.065	0.0655	0.076	0.064	0.026
Mask	0.07	0.0715	0.0675	0.062	0.0285

Table 4.10: Absolute CmAP scores for Detectors and Corruptions

By destructing the results of the table above into its individual components we can extract some conclusions for the effect of the applied on corruptions on each model. The first clear observation is that the Pixelate and Contrast corruptions are the ones causing the worst deterioration in performance, in most cases by a large margin. This result could indicate that model architecture is not the most relevant feature when it comes to robustness against similar corruptions, since these different models get affected by the same corruptions in similar ways, but the training data distribution is, since all these models were trained on the COCO dataset. This hypothesis is also supported by the fact that the corruptions which affect a detector more are common among all detectors, meaning that the Gaussian noise corruption is the third worst overall corruption for all detectors, while Shot noise is the fourth overall corruption for all detectors except YOLOv6 etc. Therefore,

based on these observations we hypothesize that training data distribution is more important when it comes to robustness against input image corruptions compared to model architecture, although this needs to be thoroughly tested with different techniques to be confirmed.

Chapter 5

Saliency Map Experiments

5.1 D-RISE Algorithm

To further expand our insight on the robustness of these object detectors we will continue by extracting the saliency maps they produce according to their predictions. To this end, we will be using the D-RISE (Detector Randomized Input Sampling for Explanation) algorithm [35], which is a black-box algorithm developed for producing saliency maps for any object detectors using their bounding box and confidence score predictions. Previous works have used an importance score that is backpropagated through the layers of the network, starting from the output and leading back to the individual pixels in the image [45], [42], [57], however, these methods rely on the knowledge of the model's architecture, rendering them unsuitable for explaining different models than the ones they were specifically designed to work on. This type of method is characterized as white-box, since it requires insight into the model's functionality, which is not always a feasible task. Furthermore, object detectors require explanations not just for the categorization of a bounding box but also for its location, which is why existing saliency methods are not suitable for this task. D-RISE is a black-box algorithm, in that it does not require any knowledge of the model's inner functioning to produce a saliency map that corresponds to its prediction. Based on the RISE algorithm [34], in this method the main idea is to measure the effect of masking random regions of the input image and then utilize the changes of the model's predictions in this image to calculate its importance. Specifically, N binary masks are generated and then used to mask the image that is input to the detector, which generates D bounding box and confidence score proposals for that image. Each of these proposals is denoted as:

$$d_i = [L_i, O_i, P_i] = [(x_1^i, y_1^i, x_2^i, y_2^i), O_i, (p_1^i, ..., p_C^i)] \tag{5.1.1}$$

where $L_i = (x_1^i, y_1^i), (x_2^i, y_2^i)$ are the bounding box corner coordinates, O_i is the probability score that this bounding box contains an object, and lastly $P_i = (p_1^i, ..., p_C^i)$ is a probability vector that contains the probability the predicted bounding box contains an object belonging to a specific class. Next, a pairwise similarity is computed between the target bounding box and confidence score and the proposals to obtain weights for each binary mask. This similarity is calculated as follows:

$$S(d_t, f(M_i \odot I)) = \max_{d_j \in f(M_i \odot I)} s(d_t, d_j) \tag{5.1.2}$$

where the similarity metric s between the target vector d_t and the current detection vector d_j is computed as the product of the three similarity aspects that need to be taken into account specifically in the case of Object Detection: bounding box similarity, confidence score similarity and class-conditional probability similarity, formally defined as:

$$s(d_t, d_j) = s_L(d_t, d_j) \cdot s_P(d_t, d_j) \cdot s_O(d_t, d_j) \tag{5.1.3}$$

where

$$s_L(d_t, d_j) = IoU(L_t, L_j), s_P(d_t, d_j) = \frac{P_t \cdot P_j}{||P_t|| \cdot ||P_j||}, s_O(d_t, d_j) = O_j \tag{5.1.4}$$

Note that in this approach the target vector can be arbitrarily defined and is not bound to be model-defined, which provides the ability to produce saliency maps for objects that the detector missed at inference time.

Lastly, the saliency map is produced as a weighted sum of the binary masks. The D-RISE algorithm is specifically tailored to object detectors since it utilizes all bounding box proposals to create the saliency map and allows us to visualize the decisions of both one-stage and two-stage object detectors since it relies solely on their predictions.

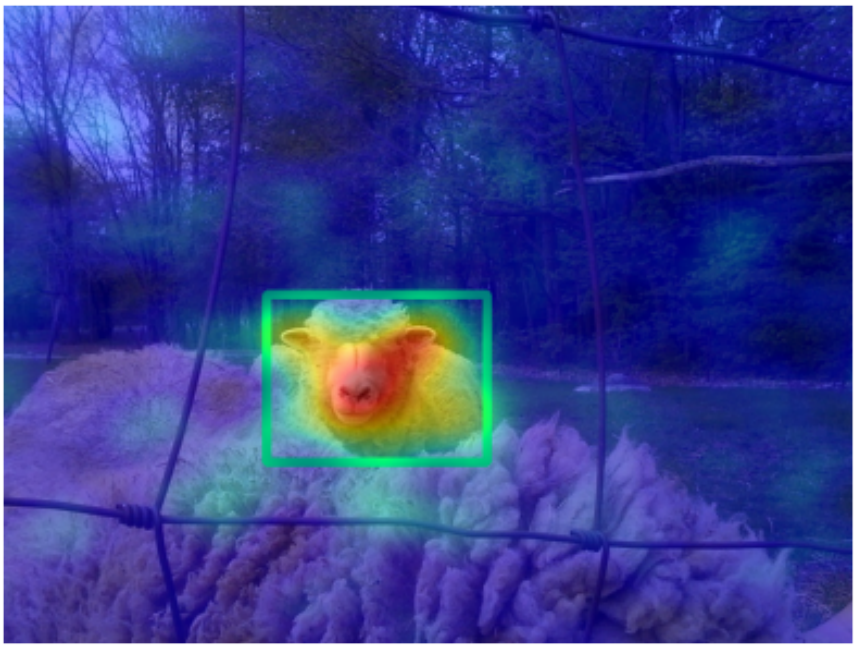
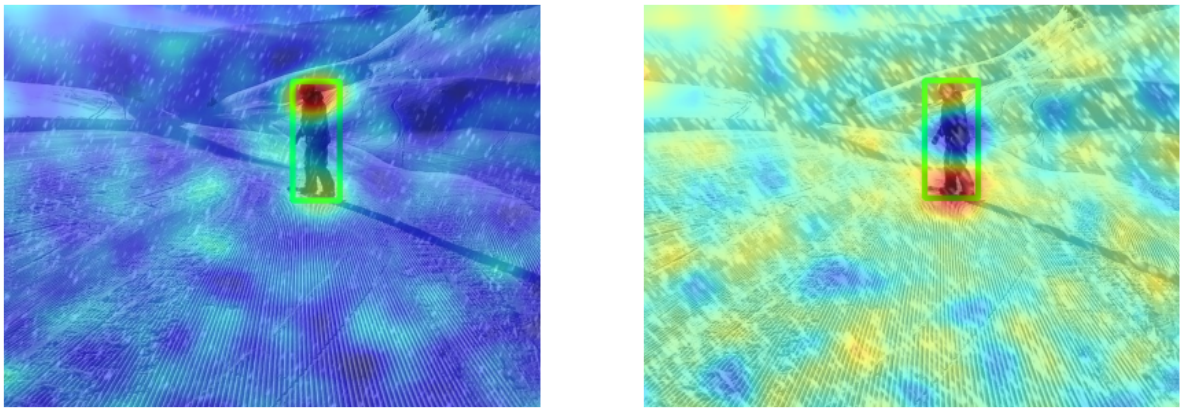


Figure 5.1.1: Saliency map produced by the D-RISE algorithm with 5000 binary masks and a probability threshold of 0.5

During our experiments we selected the default parameters of the algorithm, i.e. 1.000 binary masks for each map created (since including 5.000 masks would be too resource intensive and cost too much time) and an IoU threshold of 0.5. We will be performing our experiments on a subset of the COCO validation set since the process of extracting saliency maps is very slow and requires GPU processing, which was limited in our case.



(a) Severity Level 1

(b) Severity Level 5

Figure 5.1.2: Saliency Maps extracted for the Snow Corruption

5.2 Experiment Pipeline

Our goal is to visualize the shifts in the saliency maps produced by the D-RISE algorithm for one-stage and two-stage detectors as the severity of the corruptions we applied increases. We will be testing the YOLOv3 algorithm to represent one-stage object detectors and the Faster R-CNN algorithm to represent two-stage object detectors. Although these models are older versions of the models studied in the previous sections

the main principles of each approach to object detection remain the same, which allows us to generalize our conclusions and combine them with our previous ones. We will be using a probability threshold of 0.5 for both object detectors and 1000 binary masks to generate the saliency maps. Next, we will produce saliency maps for all images of the COCO validation set, corrupted with a subset of the corruptions used in the previous sections, for the predictions of YOLOv3 and Faster R-CNN. We selected a corruption from each category, that we determined would provide interesting visual results in order to economize on resources since the extraction of saliency maps is a costly procedure in terms of time. Specifically, we will study the Frost corruption from the Weather category, the Impulse Noise corruption, the Zoom Blur corruption and finally the Contrast corruption from the Digital category.

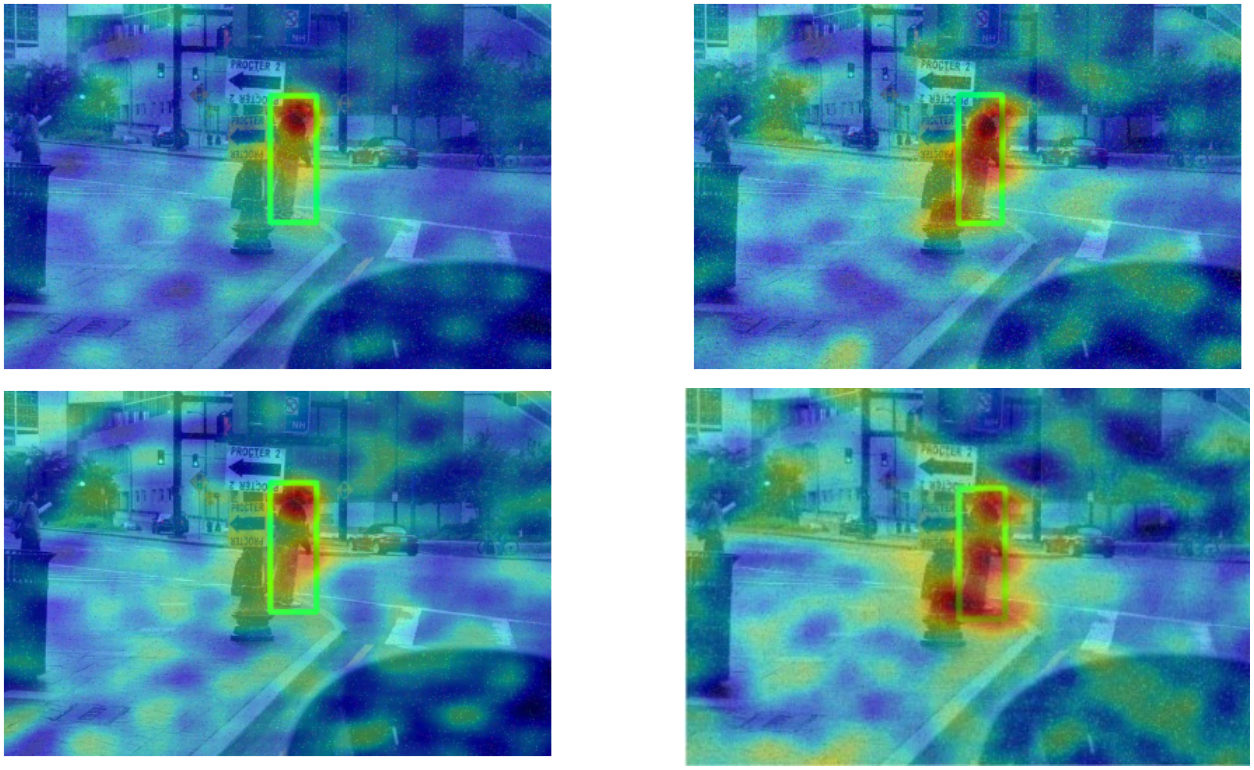


Figure 5.2.1: Saliency Maps - 4 levels of Impulse Noise Corruption

As we can tell from the saliency maps extracted above, it is quite challenging to get meaningful interpretations about the model's behaviour strictly through visual evaluation, especially when it comes to such large numbers of images and objects detected, which is why we propose three evaluation metrics to help us quantify these results and study them more thoroughly.

1. **Number of Salient Areas:** This metric represents the number of pixels in the saliency map that are marked as important to the detector's decision. A pixel is defined as salient if its value in the saliency map surpasses a specific threshold. We define that threshold as follows:

$$threshold = E[saliency\ map] + \frac{E[saliency\ map] + max(saliency\ map)}{2} \tag{5.2.1}$$

where

$$E[saliency\ map] = \frac{\sum_{i=0}^{N} saliency\ map(i)}{N} \tag{5.2.2}$$

is the mean value of the pixel values in a saliency map of N pixels total and $max(saliency\ map)$ is the maximum pixel value in the saliency map. This threshold was determined experimentally from observing the majority of saliency maps produced on our corrupted COCO validation set. This metric will help us observe which corruptions lead to an increase of salient pixels.

2. **Pixel Ratio:** This metric represents the ratio of the number of salient pixels that are found outside of a bounding box, for each object detected, over the total number of salient pixels. A pixel is considered salient if its value in the saliency map is over the threshold we proposed above. If we denote S_B as the total number of pixels of a bounding box that are salient and S_N as the total number of salient pixels we can define the pixel ratio as

$$pixel\ ratio = \frac{S_N - S_B}{S_N} \tag{5.2.3}$$

   This metric aims to quantify how important the area inside the bounding box is and how much the context outside it affects the detector's decision. A large pixel ratio value might signify that the environment of the image is more important, or that the detector gets "confused" by the corruption as its severity increases, and considers noise or other insignificant elements as objects or important context clues

3. **Box Ratio:** This metric represents the ratio of the pixels of a bounding box that are marked as salient pixels (as defined above over the total number of pixels of the bounding box. If we denote B the total number of pixels in a bounding box we can define the box ratio as

$$box\ ratio = \frac{S_B}{B} \tag{5.2.4}$$

   With this ratio we aim to determine what percentage of each bounding box is marked as salient, since it is known that the entire area inside a bounding box is not equally important to a detector's decision [35].

We will calculate these metrics for every object class contained in every image in the different versions of the corrupted COCO validation set, and then extract their average values for the most important object classes. In this context, an object class is considered important when it appears often or when it is related to important applications in object detection, such as the person, vehicle and stop sign classes.

It is important to note that only the Box Ratio and Pixel Ratio metrics are calculated for a specific bounding box, therefore for a specific object and its corresponding class, so we will calculate their values for every object in every image of the corrupted COCO validation set and then aggregate the results in order to extract class-specific conclusions. Also note that this area has not been explored in the literature so far, so these metrics are still experimental and may not definitively provide a strict visual measure for the robustness of object detectors against these corruptions. However, we believe that they are a solid foundation on which we can base the analysis of our experiments and safely draw some conclusions on the subject.

5.3 Results

We present the results of our experiments as mentioned in the previous chapter, indicatively starting with the Contrast corruption for both detectors.

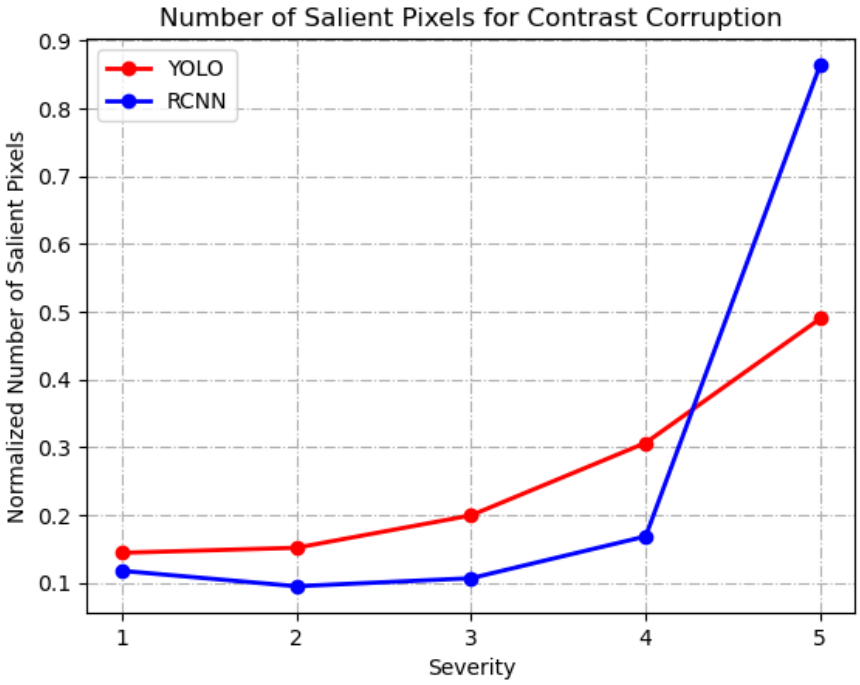


Figure 5.3.1: Number of Salient Pixels on Contrast Corruption

As expected, the number of salient pixels increases for both detectors, as the severity level increases, with YOLOv3 producing a more linear and steady increase, while Faster R-CNN remains more stable until the spike observed in the final severity level. The number of salient pixels is expected to increase since as the corruption deteriorates the image, the detector locates objects where they don't actually exist. Another possible explanation for this result is that as the corruption changes the image more and more, the underlying distribution changes, making the image different than the data the detector was trained on. Therefore, since the normal features of the images and the objects they contain change, the model believes that every part of the image is an outlier feature and therefore something that is salient, and that needs to be paid attention to.

Next, we calulcate the pixels ratios for each detector for the contrast corruption, starting with the "person" object class, which is the biggest class in the COCO dataset, containing the biggest number of instances in the dataset.

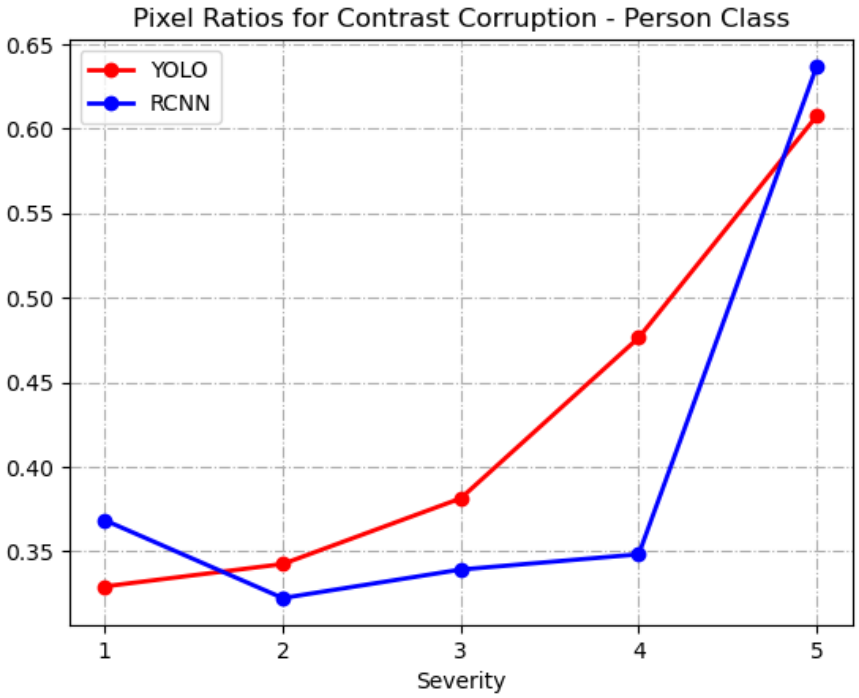


Figure 5.3.2: Pixel Ratios on Contrast Corruption for the Person Class

As we can see the pixel ratios generally increase for both detectors with the increase of corruption severity, with YOLOv3 again having an almost polynomial increase, while Faster R-CNN maintains a more steady course until the last level of corruption. Combining these results with the corresponding increase in total salient pixels, it is safe to say that as the image deteriorates, both detectors being paying more attention to outside context clues, possibly for the reasons we mentioned above, with the R-CNN model again maintaining a more steady increase until the final severity level.

Lastly, we will be calculating the box ratios for each detector, again for the contrast corruption and the person class.

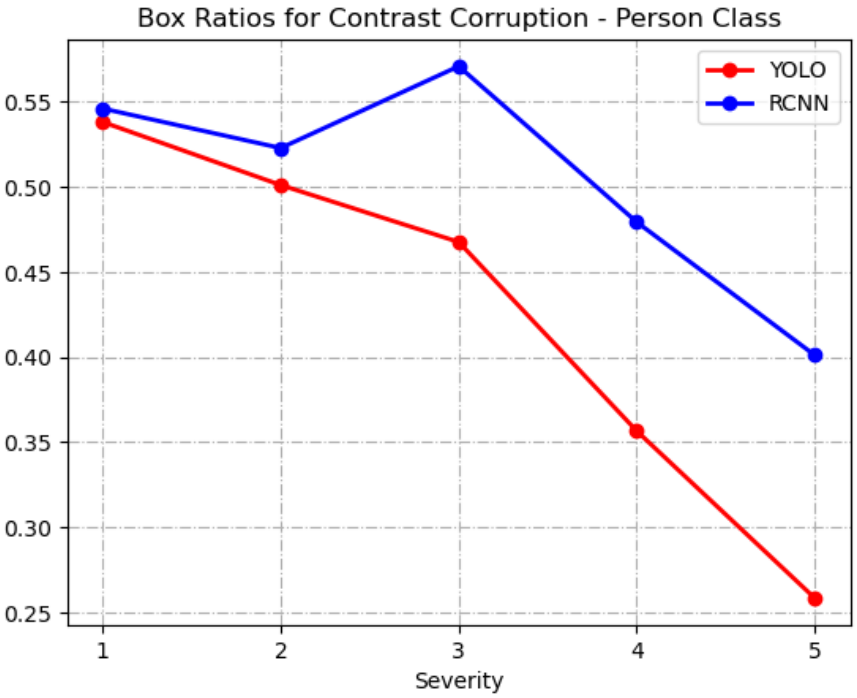


Figure 5.3.3: Box Ratios on Contrast Corruption for the Person Class

We can see that the values of this metric have a downward tendency as the severity of the corruption increases for both detectors, with Faster R-CNN again having a relatively more steady decline compared to YOLOv3. The conclusion that can be drawn from this diagram is that as the severity of the corruption increases the area inside the bounding box becomes less important to the detector, since it contains fewer and fewer salient pixels. Comparing this result with the fact that the total number of salient pixels increases, we would be justified in assuming that the surroundings of the object become more important and that the detector finds more interest in the overall environment of the image. This could mean that as the features of the image change and become different from the features the model has been trained on, it marks that environment as unknown and therefore more "interesting" and important for the detection, whereas the object class itself loses interest since its detection is encoded with far more features than the detection of its context.

Overall, the main conclusion to be made from this analysis is that, as the corruption increases in severity, the model gets "confused" and does not know where to pay attention to, which is why the area inside the bounding box which should be important becomes equally as important as any other part of the image. which explains the overall rise in pixel ratio and salient pixels and decrease in pixel ratio.

Again, these conclusions are only theoretical and cannot be directly confirmed in this line of research, however they can be viewed as hypotheses to be studied further in the future.

We will extend our analysis to the "Car" class, which is another distinct object class that appears in the dataset very often, and also plays a crucial role in the wide variety of applications of object detection models.

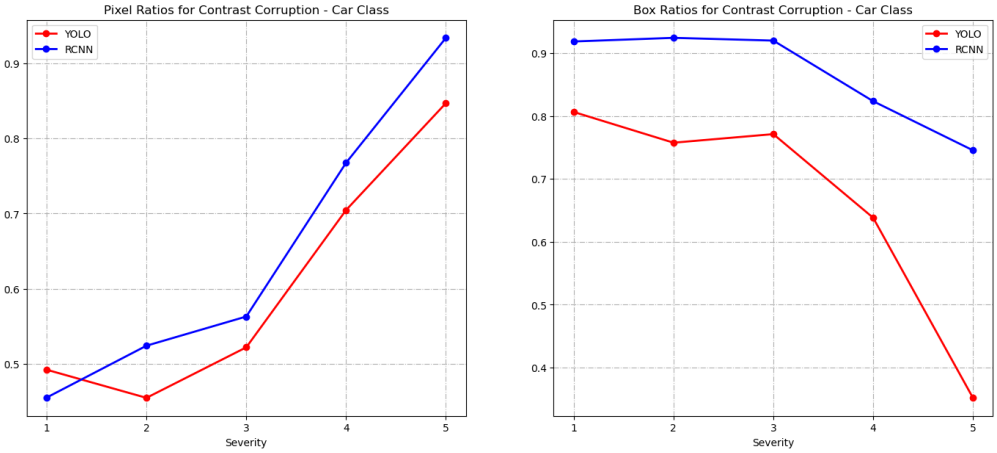


Figure 5.3.4: Saliency Metrics on Contrast Corruption - Car Class

As we can see in the produced diagram the same general principles that were present for the Person class still apply to the Car class, although the results are not identical, as expected. The pixel ratio values increase along with the increase of salient pixels, while the box ratio values decrease. Note that the number of salient pixels is a class-agnostic metric, i.e. it refers to the entire image and remains the same regardless of which class is being detected, therefore we will not be repeating it. Based on this observation we can postulate that the conclusions we draw for one class can be generalized to others, which would allow us to strictly define a framework for evaluating the robustness of these models.

To continue our analysis we will be presenting our results for a different corruption, the Frost corruption. We choose to systematically analyze these two corruptions since they contain a very important differing component that can split our analysis into two categories: how do object detectors behave against corruptions that insert new edges to the image, versus against corruptions that do not. The Frost corruption inserts an increasing amount of new edges to the image as its severity increases, so we expect the saliency analysis results to be fundamentally different from the results of the Contrast corruption, which is a "smooth" corruption, in that it does not add new elements to the image. We will be presenting the same diagrams as above for the Frost corruption.

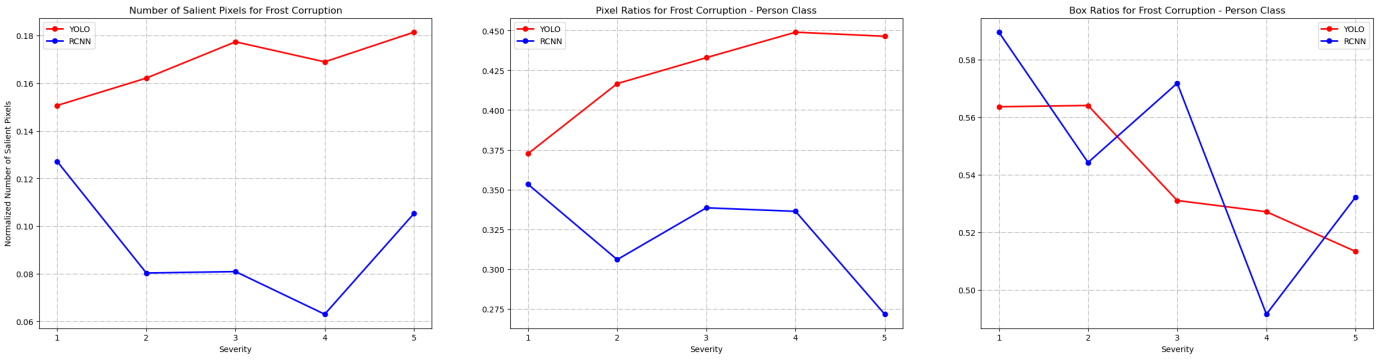


Figure 5.3.5: Saliency Metrics on Frost Corruption - Person Class

The results for this corruption are more erratic and complex, a fact which has several possible explanations. When it comes to the number of salient pixels YOLOv3 maintains the same increase as in the Contrast corruption, whereas Faster R-CNN sees an overall decrease. This result could be attributed to the fact that as the severity of the corruption increases Faster R-CNN fails to detect more and more objects. Since the saliency map is created according to the objects detected, when there is no object the saliency map includes no salient pixels.

Therefore, since the frost corruption has a more significant impact on detector performance regardless of architecture, as evidenced by our experiments in the previous chapters, the detector gets "more" confused and cannot pay attention to any significant features in particular. Ultimately, we observe the performance superiority of YOLOv3 on corruptions that add new edges and features to the image and the performance superioty of Faster R-CNN on corruptions that are smoother and affect the color scheme and other similar features of the images. This generalization stems from these results and others that are not shown in this chapter but can be found in the appendix.

Continuing with the pixel ratio metric, we see that YOLOv3 maintains the expected behaviour of increasing, however at a slower pace than the Contrast corruption, meaning that this model considers context clues more important with the decrease in image quality. On the other hand, Faster R-CNN sees an overall decrease in pixel ratios. This result, combined with the overall decrease in salient pixels could indicate that the detector does not consider context clues as important.

Lastly, when it comes to the box ratios, the metric values again maintain the same behaviour as before for the YOLOv3 detector, with a steady decrease accompanying the severity increase. Conversely, for Faster R-CNN this metric shows a very erratic behaviour, which can be attributed to poor performance, and disallows us from safely drawing any conclusions regarding its behaviour.

Next we repeat this process, this time extracting the results for the Frost corruption and the Car object class (once again the values for the total number of salient pixels remains the same).

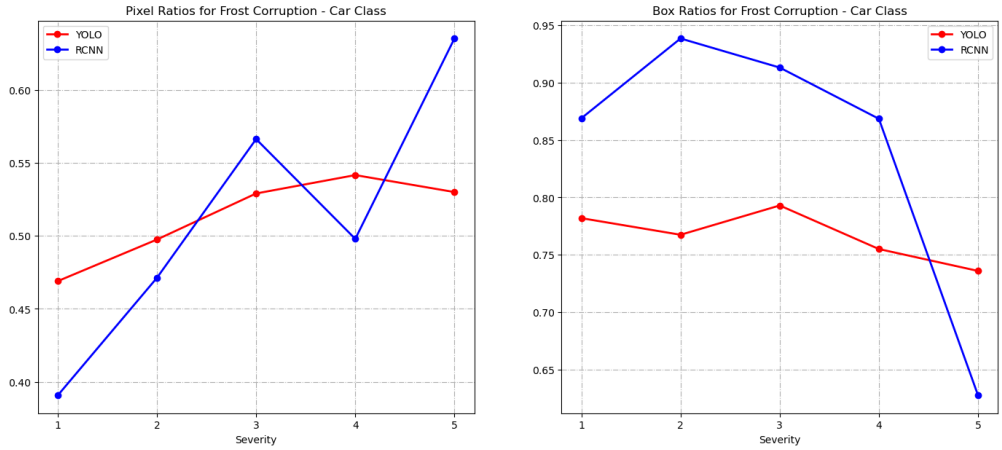


Figure 5.3.6: Saliency Metrics on Frost Corruption - Car Class

Similarly to the contrast corruption, we see that the pixel ratio metric diagram closely follows the pixel ratio diagram for the Person class, for both detectors, which reinforces our assumption that the same corruption usually affects the detector's ability to detect different object classes in the same way. The first change is observed in the box ratio diagram, where YOLOv3 behaves in a similar way as it did with the Person class, with an overall decrease, however Faster R-CNN also shows a clear decrease, which clashes with the results of the Person class. This fact could be attributed to the detector's deteriorating performance, where fewer instances of the Person class were detected due to the corruption in some severity levels and then more in other levels, which lead to the erratic behaviour noticed in the previous diagram, however with the Car class this might not have been the case, which leads to this expected behaviour. So once again, as the corruption deteriorates the input image the content inside the bounding box becomes less important for the decision but at the same time context clues do not become more important as evidenced by the simultaneous decrease in pixel ratio values. These results indicate that as the corruption gets more severe, the detector cannot find important features to focus on as the image gets more cluttered, especially in the cases of corruptions such as this that add new elements and edges to the image, which leads to failure to detect the object and the resulting deteriorating performance.

We also provide the results for the rest of the corruptions which are similar to the ones we have analyzed thus far.

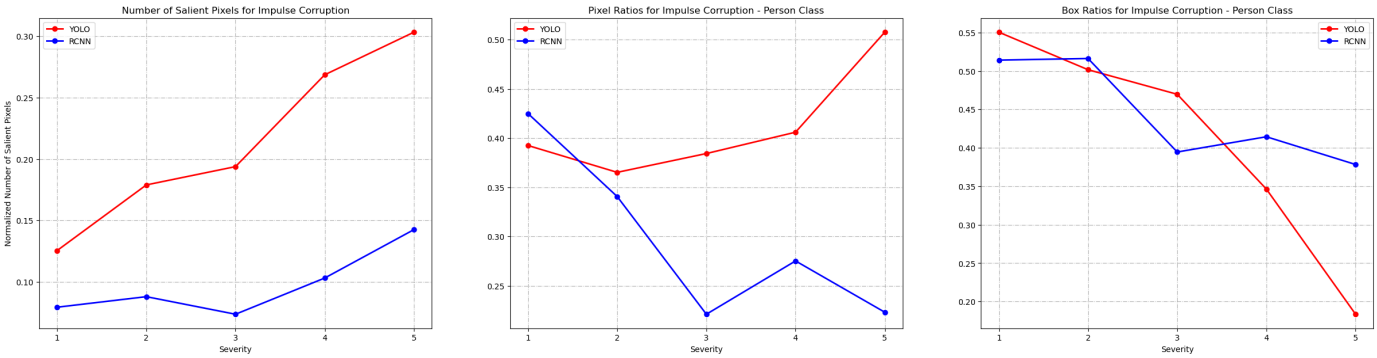


Figure 5.3.7: Saliency Metrics on Impulse Corruption - Person Class

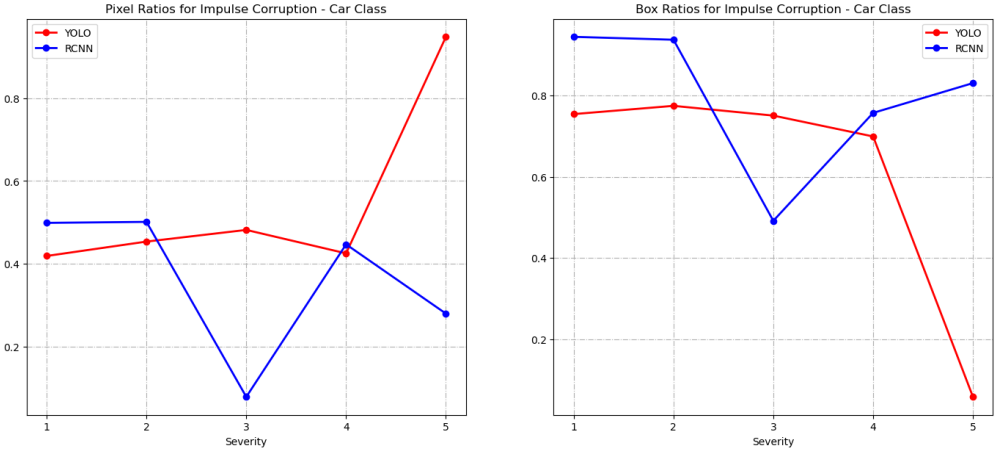


Figure 5.3.8: Saliency Metrics on Impulse Corruption - Car Class

As we can see YOLOv3's behaviour under the Impulse Noise corruption follows the Contrast corruption, with the increase of salient pixels and pixel ratio accompanied by the decrease of box ratio. That seems to not be the base with Faster R-CNN, however we can attribute that to an inability to detect most object instances under this corruption since we can see that the number of salient pixels is very low compared to YOLOv3 and the pixel ratio metric is very close to 0. Nevertheless, the number of salient pixels and box ratio seem to follow YOLOv3. We can attribute this similarity of performance of the models on Impulse Noise and Contrast on the overall similarity of the corruptions, which, as was mentioned in previous chapters, boils down to the fact that both they disrupt the objects' edges by making them appear fused, blurred and subdued.

Lastly, we provide the results for the Zoom Blur corruption.

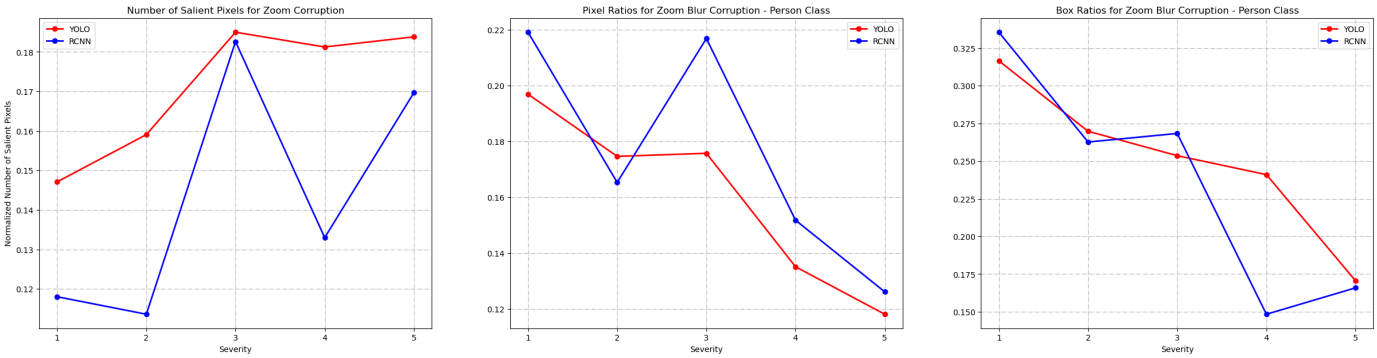


Figure 5.3.9: Saliency Metrics on Zoom Corruption - Person Class

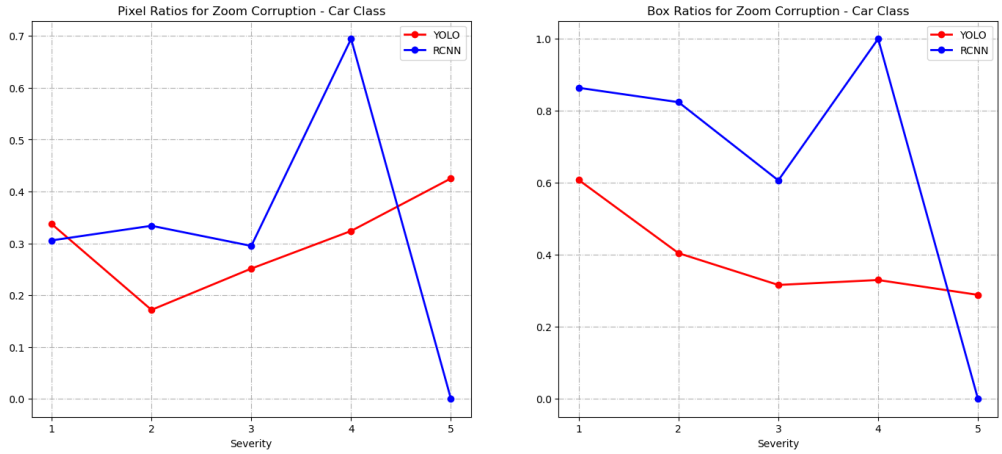


Figure 5.3.10: Saliency Metrics on Zoom Corruption - Car Class

Both detectors' performance is very erratic when performing under this corruption, with some patterns briefly appearing in the box ratio curves for both classes that match our conclusion about the Impulse and Contrast corruptions. However, taking into account the poor performance of the detectors in the previous chapter, where Zoom blur caused some of the worst results in terms of absolute mAP score in all detectors, we can attribute this phenomenon to the models' poor performance on this corruption, which leads to incomplete results.

The results for the rest of the object classes were quite limited due to the fact that their instances are much fewer than those of the two classes analyzed in this chapter, especially given we only worked on a subset of the COCO validation set. Therefore, we will not be providing the results of these experiments in this section.

Chapter 6

Synonpsis, Conclusions and Future Steps

6.1 Synopsis

In this work we experimented with the most state of the art one-stage and two-stage object detectors in order to systematically analyze their behaviour under several types of image corruptions. Our goal was to evaluate the robustness of the best current models against common perturbations, underline possible differences among different models of the same category as well as possible differences among models whose architectures differ greatly, and offer interpretations that could push forward research in this domain. We started by creating 18 datasets stemming from the MS COCO validation set, each of which contains 25.000 images: the original 5.000 images of the dataset with 5 increasing levels of each corruption applied. Next we applied the most modern object detection models from the YOLO and the R-CNN family and observed the deterioration in their performance, documenting our observations and conclusions along the way. Furthermore, we extracted the corresponding saliency maps for the corruptions that yielded the biggest deterioration in performance and proposed three new metrics to quantify the analysis performed on them that accompanied our qualitative analysis.

6.2 Conclusions

The most fundamental conclusion that can be drawn from our experiments is that, despite the constant increase in object detection performance by the new models that are being produced, robust object detection still remains wide open, with the newer models not necessarily surpassing the old ones when it comes to robustness against input image corruption. We found that, although the newer version of the YOLO algorithm could overall perform better on corrupted inputs, their overall decline in performance as the severity of the corruption increased was not any slower when it came to the newest model. Comparing them to the much older Mask R-CNN model, we found that they were even less robust, which leads us to yet another interesting conclusion: that is that one-stage object detectors do not only sacrifice accuracy in order to gain inference speed but also sacrifice robustness. Furthermore, we observed that the more complex models in terms of size (number of parameters etc.) were not strictly more robust than their smaller equivalents, even though their overall performance was superior. These results lead us to believe that the solution to creating more robust object detectors is not adding more complexity to the network architecture, but other techniques such as data augmentation, pretraining techniques etc. Our analysis using saliency maps also offered some interesting observations, such as the fact that the nature of the corruption plays a decisive role in the way the detector makes decisions, with corruptions that insert new edges to an image having a different impact compared to smoother corruptions. Specifically, our results showed that smooth corruptions cause the detector to pay more attention to contextual clues and the overall environment of the image, while the corruptions that inserted new edges produced more erratic results in terms of saliency map analysis, probably due to the fact that they caused a more serious deterioration in accuracy and therefore the amount of object detected was reduced significantly. However, the most severe results both for Object Detection and Saliency Map analysis were caused by corruptions that affected the existing image edges, by blurring or pixelating them thus making

the objects harder to distinguish from the background or from each other. This leads us to the important conclusion that the edges of the objects in the image are perhaps the most important features when it comes to recognizing and classifying them. Furthermore, based on the range of object classes we studied, we found that the nature of the corruption does not affect the decision making process of the detector differently for each class, but the whole process can be considered universal.

Concluding, we can state that model performance against image corruptions can be modeled and later studied in order to develop techniques targeted in improving robustness. We were able to provide some initial steps towards this effort by recognizing patterns that appeared in the data, and hopefully our experiments can be used as a starting point for even more extensive research in the field of robustness.

6.3 Future Steps

Robustness in ML, and more specifically in Computer Vision is a wide and engaging field gaining more and more importance as these models claim a more prominent position in modern life every day. Ongoing research on this field is remarkably active, and we wish to contribute to those efforts by proposing some interesting future steps that are based on this current work. To this end we propose:

- The expansion of the set of corruptions that are applied to the input images of each model to include an ever wider spectrum of phenomena, as well as the refinement of the corruptions that already available to simulate these phenomena even more accurately and therefore simulate the response of the models tested against them as accurately as possible. One possible method to achieve this goal would be to include the use of GANs to generate more convincing effects and also add new ones, such as a realistic nighttime effect, a physics-accurate shadow effect, an added smoke effect etc. In general, the wider the range of conditions that we are able to simulate, the more comprehensive and analytical our study will be, which will allow us to create more robust and reliable models.
- The addition of an extensive occlusion robustness study on these detectors. Occlusion is a major factor in the deterioration of performance of most computer vision models, since it poses some unique challenges on most related tasks, therefore carrying out a comprehensive study of occlusion on state of the art object detection models that could aid in developing techniques that overcome this issue, is a future research direction that shows great promise.
- The extension of our experiments to video input. Particularly the YOLO detectors are extremely popular when it comes to object detection in video, or even real-time object detection. Therefore, it would be of great interest to observe their performance on corrupted video input, and determine whether the principles of robustness from object detection on images carry over to object detection on video. Occlusion study can also be included in object detection with video input. Especially for applications like autonomous driving and navigation, this type of study is crucial in the real-world deployment of these models.
- The extension of our saliency map analysis to include different object classes, corruptions and a larger dataset. As mentioned in the corresponding chapter, we performed our experiments on a subset of the COCO validation set images and object classes that are most important to certain applications. However, given freedom of computational resources, these experiments can be repeated for all object classes and images, in order to validate the results that have been extracted thus far and perhaps generalize them.
- The implementation of this experiment framework on different models outside the scope of Computer Vision. An interesting future direction is the experimentation on Visual Commonsense Reasoning models, which combine the domains of Computer Vision and Natural Language Processing to create algorithms that, given a visual and a written contextual input, can draw logical conclusions and produce interpretations about a scene, what is currently occurring in it or what might take place in the future. This line of research remains relatively unexplored, therefore testing the robustness of these newer models by adding corrupted image inputs could yield some interesting results, even for the more basic Computer Vision principles they are founded on.

Chapter 7

Bibliography

[1] S. F. Altindis et al. *Benchmarking the Robustness of Instance Segmentation Models*. 2021. DOI: 10.48550/ARXIV.2109.01123. URL: https://arxiv.org/abs/2109.01123.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. DOI: 10.48550/ARXIV.2004.10934. URL: https://arxiv.org/abs/2004.10934.

[3] N. Carlini and D. Wagner. *Towards Evaluating the Robustness of Neural Networks*. 2016. DOI: 10.48550/ARXIV.1608.04644. URL: https://arxiv.org/abs/1608.04644.

[4] J. Chen et al. *Robust Attribution Regularization*. 2019. DOI: 10.48550/ARXIV.1905.09957. URL: https://arxiv.org/abs/1905.09957.

[5] X. Chen et al. *Robust and Accurate Object Detection via Adversarial Learning*. 2021. DOI: 10.48550/ARXIV.2103.13886. URL: https://arxiv.org/abs/2103.13886.

[6] J. I. Choi and Q. Tian. *Adversarial Attack and Defense of YOLO Detectors in Autonomous Driving Scenarios*. 2022. DOI: 10.48550/ARXIV.2202.04781. URL: https://arxiv.org/abs/2202.04781.

[7] *COCO - Common Objects in Context*. 2014. URL: https://cocodataset.org/#home.

[8] M. Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge." In: *Int. J. Comput. Vis.* 88.2 (2010), pp. 303–338. URL: http://dblp.uni-trier.de/db/journals/ijcv/ijcv88.html#EveringhamGWWZ10.

[9] R. C. Fong and A. Vedaldi. "Interpretable Explanations of Black Boxes by Meaningful Perturbation". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2017. DOI: 10.1109/iccv.2017.371. URL: https://doi.org/10.1109%2Ficcv.2017.371.

[10] J. Fritsch, T. Kuehnl, and A. Geiger. "A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms". In: *International Conference on Intelligent Transportation Systems (ITSC)*. 2013.

[11] H. Fujiyoshi, T. Hirakawa, and T. Yamashita. "Deep learning-based image recognition for autonomous driving". In: *IATSS research* 43.4 (2019), pp. 244–252.

[12] R. Geirhos et al. *Generalisation in humans and deep neural networks*. 2018. DOI: 10.48550/ARXIV.1808.08750. URL: https://arxiv.org/abs/1808.08750.

[13] R. Girshick. *Fast R-CNN*. 2015. DOI: 10.48550/ARXIV.1504.08083. URL: https://arxiv.org/abs/1504.08083.

[14] R. Girshick et al. *Rich feature hierarchies for accurate object detection and semantic segmentation*. 2013. DOI: 10.48550/ARXIV.1311.2524. URL: https://arxiv.org/abs/1311.2524.

[15] Y. Gong et al. *SSAST: Self-Supervised Audio Spectrogram Transformer*. 2021. DOI: 10.48550/ARXIV.2110.09784. URL: https://arxiv.org/abs/2110.09784.

[16] C. Guo et al. *Countering Adversarial Images using Input Transformations*. 2017. DOI: 10.48550/ARXIV.1711.00117. URL: https://arxiv.org/abs/1711.00117.

[17] K. He et al. *Mask R-CNN*. 2017. DOI: 10.48550/ARXIV.1703.06870. URL: https://arxiv.org/abs/1703.06870.

[18] D. Hendrycks and T. G. Dietterich. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *CoRR* abs/1903.12261 (2019). arXiv: 1903.12261. URL: http://arxiv.org/abs/1903.12261.

[19] M. Huh, P. Agrawal, and A. A. Efros. *What makes ImageNet good for transfer learning?* 2016. DOI: 10.48550/ARXIV.1608.08614. URL: https://arxiv.org/abs/1608.08614.

[20] A. Ignatiev, N. Narodytska, and J. Marques-Silva. "On Relating Explanations and Adversarial Examples". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/7392ea4ca76ad2fb4c9c3b6a5c6e31e3-Paper.pdf.

[21] C. Kamann and C. Rother. "Benchmarking the Robustness of Semantic Segmentation Models". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2020. DOI: 10.1109/cvpr42600.2020.00885. URL: https://doi.org/10.1109%2Fcvpr42600.2020.00885.

[22] C. Li et al. *YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*. 2022. DOI: 10.48550/ARXIV.2209.02976. URL: https://arxiv.org/abs/2209.02976.

[23] J. Li et al. *Learning to Learn from Noisy Labeled Data*. 2018. DOI: 10.48550/ARXIV.1812.05214. URL: https://arxiv.org/abs/1812.05214.

[24] T.-Y. Lin et al. *Focal Loss for Dense Object Detection*. 2017. DOI: 10.48550/ARXIV.1708.02002. URL: https://arxiv.org/abs/1708.02002.

[25] W. Liu et al. "SSD: Single Shot MultiBox Detector". In: *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 21–37. DOI: 10.1007/978-3-319-46448-0_2. URL: https://doi.org/10.1007%2F978-3-319-46448-0_2.

[26] J. Lu, H. Sibai, and E. Fabry. *Adversarial Examples that Fool Detectors*. 2017. DOI: 10.48550/ARXIV.1712.02494. URL: https://arxiv.org/abs/1712.02494.

[27] M. Lymperaiou, K. Thomas, and G. Stamou. *Fine-Grained ImageNet Classification in the Wild*. 2023. DOI: 10.48550/ARXIV.2303.02400. URL: https://arxiv.org/abs/2303.02400.

[28] N. G. Maity and S. Das. "Machine learning for improved diagnosis and prognosis in healthcare". In: *2017 IEEE aerospace conference*. IEEE. 2017, pp. 1–9.

[29] P. Mangla, V. Singh, and V. N. Balasubramanian. *On Saliency Maps and Adversarial Robustness*. 2020. DOI: 10.48550/ARXIV.2006.07828. URL: https://arxiv.org/abs/2006.07828.

[30] G. Mattolin et al. *ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing*. 2022. DOI: 10.48550/ARXIV.2210.11539. URL: https://arxiv.org/abs/2210.11539.

[31] D. Meng and H. Chen. *MagNet: a Two-Pronged Defense against Adversarial Examples*. 2017. DOI: 10.48550/ARXIV.1705.09064. URL: https://arxiv.org/abs/1705.09064.

[32] J. H. Metzen et al. *On Detecting Adversarial Perturbations*. 2017. DOI: 10.48550/ARXIV.1702.04267. URL: https://arxiv.org/abs/1702.04267.

[33] C. Michaelis et al. *Benchmarking Robustness in Object Detection: Autonomous Driving when Winter is Coming*. 2019. DOI: 10.48550/ARXIV.1907.07484. URL: https://arxiv.org/abs/1907.07484.

[34] V. Petsiuk, A. Das, and K. Saenko. *RISE: Randomized Input Sampling for Explanation of Black-box Models*. 2018. DOI: 10.48550/ARXIV.1806.07421. URL: https://arxiv.org/abs/1806.07421.

[35] V. Petsiuk et al. *Black-box Explanation of Object Detectors via Saliency Maps*. 2020. DOI: 10.48550/ARXIV.2006.03204. URL: https://arxiv.org/abs/2006.03204.

[36] A. Prakash et al. *Deflecting Adversarial Attacks with Pixel Deflection*. 2018. DOI: 10.48550/ARXIV.1801.08926. URL: https://arxiv.org/abs/1801.08926.

[37] B. Recht et al. *Do ImageNet Classifiers Generalize to ImageNet?* 2019. DOI: 10.48550/ARXIV.1902.10811. URL: https://arxiv.org/abs/1902.10811.

[38] J. Redmon and A. Farhadi. *YOLO9000: Better, Faster, Stronger*. 2016. DOI: 10.48550/ARXIV.1612.08242. URL: https://arxiv.org/abs/1612.08242.

[39] J. Redmon and A. Farhadi. *YOLOv3: An Incremental Improvement*. 2018. DOI: 10.48550/ARXIV.1804.02767. URL: https://arxiv.org/abs/1804.02767.

[40] J. Redmon et al. *You Only Look Once: Unified, Real-Time Object Detection*. 2015. DOI: 10.48550/ARXIV.1506.02640. URL: https://arxiv.org/abs/1506.02640.

[41] S. Ren et al. *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. 2015. DOI: 10.48550/ARXIV.1506.01497. URL: https://arxiv.org/abs/1506.01497.

[42] B. Sebastian et al. *On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation*. 2015. DOI: https://doi.org/10.1371/journal.pone.0130140. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0130140.

[43] R. R. Selvaraju et al. “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization”. In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. DOI: 10.1007/s11263-019-01228-7. URL: https://doi.org/10.1007%2Fs11263-019-01228-7.

[44] K. Simonyan, A. Vedaldi, and A. Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2013. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.

[45] K. Simonyan, A. Vedaldi, and A. Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2013. DOI: 10.48550/ARXIV.1312.6034. URL: https://arxiv.org/abs/1312.6034.

[46] J. Su, D. V. Vargas, and K. Sakurai. “One Pixel Attack for Fooling Deep Neural Networks”. In: *IEEE Transactions on Evolutionary Computation* 23.5 (Oct. 2019), pp. 828–841. DOI: 10.1109/tevc.2019.2890858. URL: https://doi.org/10.1109%2Ftevc.2019.2890858.

[47] M. Tan, R. Pang, and Q. V. Le. “EfficientDet: Scalable and Efficient Object Detection”. In: (2019). DOI: 10.48550/ARXIV.1911.09070. URL: https://arxiv.org/abs/1911.09070.

[48] R. Taori et al. *Measuring Robustness to Natural Distribution Shifts in Image Classification*. 2020. DOI: 10.48550/ARXIV.2007.00644. URL: https://arxiv.org/abs/2007.00644.

[49] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. *YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*. 2022. DOI: 10.48550/ARXIV.2207.02696. URL: https://arxiv.org/abs/2207.02696.

[50] B. Wilson, J. Hoffman, and J. Morgenstern. “Predictive Inequity in Object Detection”. In: *CoRR* abs/1902.11097 (2019). arXiv: 1902.11097. URL: http://arxiv.org/abs/1902.11097.

[51] C. Xie et al. *Mitigating Adversarial Effects Through Randomization*. 2017. DOI: 10.48550/ARXIV.1711.01991. URL: https://arxiv.org/abs/1711.01991.

[52] D. Y. Yang et al. “Building Towards "Invisible Cloak": Robust Physical Adversarial Attack on YOLO Object Detector”. In: *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. 2018, pp. 368–374. DOI: 10.1109/UEMCON.2018.8796670.

[53] *YOLOv5*. https://github.com/ultralytics/yolov5.

[54] *YOLOv8*. https://github.com/ultralytics/ultralytics.

[55] W. Yue et al. “Machine learning with applications in breast cancer diagnosis and prognosis”. In: *Designs* 2.2 (2018), p. 13.

[56] M. D. Zeiler and R. Fergus. *Visualizing and Understanding Convolutional Networks*. 2013. DOI: 10.48550/ARXIV.1311.2901. URL: https://arxiv.org/abs/1311.2901.

[57] J. Zhang et al. *Top-down Neural Attention by Excitation Backprop*. 2016. DOI: 10.48550/ARXIV.1608.00507. URL: https://arxiv.org/abs/1608.00507.

Chapter 8

Appendix

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.252	0.267	0.399	0.246	0.226
Frost	0.297	0.308	0.438	0.292	0.262
Fog	0.32	0.332	0.453	0.315	0.279
Brightness	0.357	0.362	0.48	0.24	0.345
Darken	0.349	0.356	0.468	0.347	0.315
Rain	0.339	0.345	0.468	0.336	0.325
Gauss	0.27	0.302	0.413	0.259	0.264
Impulse	0.224	0.28	0.366	0.221	0.194
Shot	0.27	0.306	0.412	0.262	0.263
Defocus	0.305	0.305	0.415	0.302	0.254
Zoom	0.131	0.131	0.208	0.127	0.108
Motion	0.289	0.306	0.409	0.289	0.269
Jpeg	0.282	0.313	0.347	0.277	0.263
Contrast	0.317	0.329	0.454	0.312	0.279
Pixelate	0.266	0.342	0.376	0.307	0.262
Elastic	0.298	0.318	0.418	0.308	0.278
Mask	0.274	0.251	0.404	0.257	0.255

Table 8.1: mAP Scores for **Small Detectors - Severity Level 1**

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.445	0.441	0.461	0.44
Frost	0.476	0.424	0.489	0.47
Fog	0.491	0.483	0.499	0.483
Brightness	0.524	0.506	0.532	0.512
Darken	0.508	0.496	0.517	0.5
Rain	0.513	0.477	0.522	0.502
Gauss	0.464	0.457	0.479	0.46
Impulse	0.446	0.433	0.459	0.424
Shot	0.464	0.46	0.48	0.458
Defocus	0.433	0.432	0.44	0.461
Zoom	0.213	0.222	0.216	0.231
Motion	0.436	0.438	0.448	0.447
Jpeg	0.447	0.441	0.452	0.377
Contrast	0.49	0.485	0.5	0.484
Pixelate	0.455	0.466	0.455	0.429
Elastic	0.437	0.437	0.454	0.447
Mask	0.482	0.454	0.501	0.451

Table 8.2: mAP Scores for **Large Detectors - Severity Level 1**

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.171	0.181	0.343	0.155	0.145
Frost	0.23	0.244	0.395	0.223	0.191
Fog	0.304	0.318	0.443	0.3	0.254
Brightness	0.347	0.353	0.47	0.341	0.329
Darken	0.341	0.349	0.461	0.338	0.315
Rain	0.317	0.327	0.456	0.315	0.302
Gauss	0.206	0.253	0.359	0.187	0.207
Impulse	0.162	0.225	0.317	0.151	0.152
Shot	0.205	0.252	0.353	0.114	0.207
Defocus	0.27	0.266	0.391	0.268	0.209
Zoom	0.086	0.085	0.145	0.0825	0.07
Motion	0.218	0.242	0.339	0.218	0.205
Jpeg	0.219	0	0.273	0.221	0.21
Contrast	0.289	0.307	0.441	0.284	0.239
Pixelate	0.215	0.332	0.312	0.293	0.228
Elastic	0.26	0.281	0.372	0.271	0.245
Mask	0.274	0.196	0.34	0.19	0.213

Table 8.3: mAP Scores for **Small Detectors - Severity Level 2**

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.386	0.425	0.395	0.397
Frost	0.476	0.424	0.44	0.43
Fog	0.479	0.476	0.489	0.471
Brightness	0.515	0.499	0.523	0.512
Darken	0.499	0.49	0.507	0.492
Rain	0.5	0.477	0.511	0.492
Gauss	0.417	0.438	0.438	0.413
Impulse	0.402	0.417	0.424	0.38
Shot	0.414	0.418	0.437	0.41
Defocus	0.386	0.432	0.371	0.423
Zoom	0.11	0.222	0.152	0.161
Motion	0.36	0.372	0.371	0.383
Jpeg	0.4	0.394	0.406	0.294
Contrast	0.476	0.474	0.484	0.469
Pixelate	0.426	0.442	0.42	0.374
Elastic	0.387	0.389	0.403	0.4
Mask	0.447	0.392	0.46	0.41

Table 8.4: mAP Scores for **Large Detectors - Severity Level 2**

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.169	0.18	0.325	0.15	0.142
Frost	0.192	0.205	0.364	0.223	0.154
Fog	0.289	0.303	0.436	0.283	0.232
Brightness	0.334	0.343	0.458	0.328	0.305
Darken	0.331	0.337	0.449	0.326	0.271
Rain	0.264	0.279	0.424	0.259	0.244
Gauss	0.127	0.176	0.288	0.101	0.142
Impulse	0.119	0.182	0.277	0.101	0.129
Shot	0.136	0.184	0.284	0.114	0.144
Defocus	0.195	0.189	0.287	0.188	0.062
Zoom	0.063	0.062	0.11	0.0618	0.049
Motion	0.14	0.158	0.255	0.137	0.133
Jpeg	0.179	0.242	0.21	0.187	0.173
Contrast	0.234	0.265	0.415	0.23	0.173
Pixelate	0.093	0.252	0.175	0.152	0.133
Elastic	0.21	0.226	0.306	0.218	0.194
Mask	0.183	0.14	0.34	0.151	0.191

Table 8.5: mAP Scores for **Small Detectors - Severity Level 3**

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.366	0.371	0.38	0.368
Frost	0.4	0.391	0.407	0.398
Fog	0.468	0.465	0.478	0.463
Brightness	0.504	0.489	0.512	0.491
Darken	0.485	0.478	0.492	0.479
Rain	0.465	0.46	0.477	0.431
Gauss	0.352	0.359	0.377	0.349
Impulse	0.334	0.36	0.391	0.34
Shot	0.351	0.361	0.378	0.348
Defocus	0.301	0.312	0.301	0.338
Zoom	0.11	0.12	0.115	0.125
Motion	0.263	0.283	0.277	0.294
Jpeg	0.369	0.361	0.38	0.252
Contrast	0.442	0.446	0.453	0.442
Pixelate	0.312	0.343	0.311	0.256
Elastic	0.317	0.322	0.33	0.334
Mask	0.411	0.343	0.422	0.391

Table 8.6: mAP Scores for **Large Detectors - Severity Level 3**

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.128	0.134	0.278	0.114	0.098
Frost	0.18	0.196	0.361	0.175	0.146
Fog	0.287	0.3	0.432	0.282	0.231
Brightness	0.317	0.329	0.44	0.31	0.274
Darken	0.301	0.312	0.423	0.293	0.218
Rain	0.192	0.218	0.39	0.188	0.178
Gauss	0.059	0.097	0.195	0.0408	0.075
Impulse	0.05	0.089	0.174	0.0356	0.069
Shot	0.054	0.091	0.166	0.0403	0.066
Defocus	0.132	0.13	0.22	0.123	0.093
Zoom	0.045	0.044	0.081	0.0433	0.034
Motion	0.08	0.091	0.174	0.077	0.075
Jpeg	0.103	0.164	0.107	0.114	0.102
Contrast	0.112	0.16	0.336	0.114	0.06
Pixelate	0.045	0.124	0.091	0.066	0.065
Elastic	0.174	0.19	0.259	0.186	0.162
Mask	0.143	0.11	0.315	0.122	0.155

Table 8.7: mAP Scores for **Small Detectors - Severity Level 4**

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.305	0.322	0.322	0.321
Frost	0.396	0.39	0.405	0.396
Fog	0.464	0.46	0.474	0.459
Brightness	0.487	0.476	0.495	0.475
Darken	0.453	0.451	0.461	0.452
Rain	0.438	0.436	0.449	0.428
Gauss	0.262	0.282	0.301	0.259
Impulse	0.266	0.276	0.307	0.245
Shot	0.243	0.268	0.282	0.236
Defocus	0.231	0.243	0.239	0.258
Zoom	0.082	0.089	0.084	0.093
Motion	0.171	0.192	0.184	0.203
Jpeg	0.272	0.268	0.293	0.146
Contrast	0.337	0.363	0.36	0.361
Pixelate	0.183	0.228	0.187	0.168
Elastic	0.268	0.273	0.278	0.285
Mask	0.36	0.305	0.374	0.357

Table 8.8: mAP Scores for **Large Detectors - Severity Level 4**

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.12	0.135	0.283	0.11	0.091
Frost	0.157	0.17	0.34	0.151	0.112
Fog	0.261	0.278	0.409	0.258	0.206
Brightness	0.294	0.31	0.418	0.29	0.24
Darken	0.211	0.231	0.347	0.2	0.115
Rain	0.185	0.207	0.383	0.169	0.171
Gauss	0.015	0.031	0.082	0.0105	0.026
Impulse	0.014	0.031	0.079	0.101	0.029
Shot	0.023	0.048	0.089	0.169	0.034
Defocus	0.085	0.087	0.162	0.0771	0.062
Zoom	0.034	0.035	0.066	0.0353	0.024
Motion	0.054	0.061	0.132	0.0523	0.05
Jpeg	0.051	0.1	0.053	0.187	0.051
Contrast	0.024	0.052	0.203	0.23	0.008
Pixelate	0.029	0.076	0.037	0.0442	0.029
Elastic	0.135	0.146	0.203	0.145	0.121
Mask	0.078	0.059	0.224	0.0646	0.085

Table 8.9: mAP Scores for **Small Detectors - Severity Level 5**

Detector	YOLOv5x	YOLOv6l	YOLOv7E6	YOLOv8x
Snow	0.313	0.331	0.324	0.331
Frost	0.366	0.364	0.38	0.371
Fog	0.445	0.441	0.453	0.435
Brightness	0.466	0.458	0.475	0.455
Darken	0.358	0.369	0.372	0.373
Rain	0.422	0.426	0.432	0.414
Gauss	0.151	0.179	0.195	0.141
Impulse	0.16	0.185	0.212	0.144
Shot	0.168	0.201	0.089	0.159
Defocus	0.171	0.183	0.181	0.188
Zoom	0.069	0.07	0.069	0.0751
Motion	0.125	0.143	0.136	0.146
Jpeg	0.15	0.177	0.191	0.0764
Contrast	0.166	0.216	0.216	0.221
Pixelate	0.091	0.127	0.094	0.0764
Elastic	0.208	0.215	0.213	0.226
Mask	0.256	0.218	0.275	0.27

Table 8.10: mAP Scores for **Large Detectors - Severity Level 5**

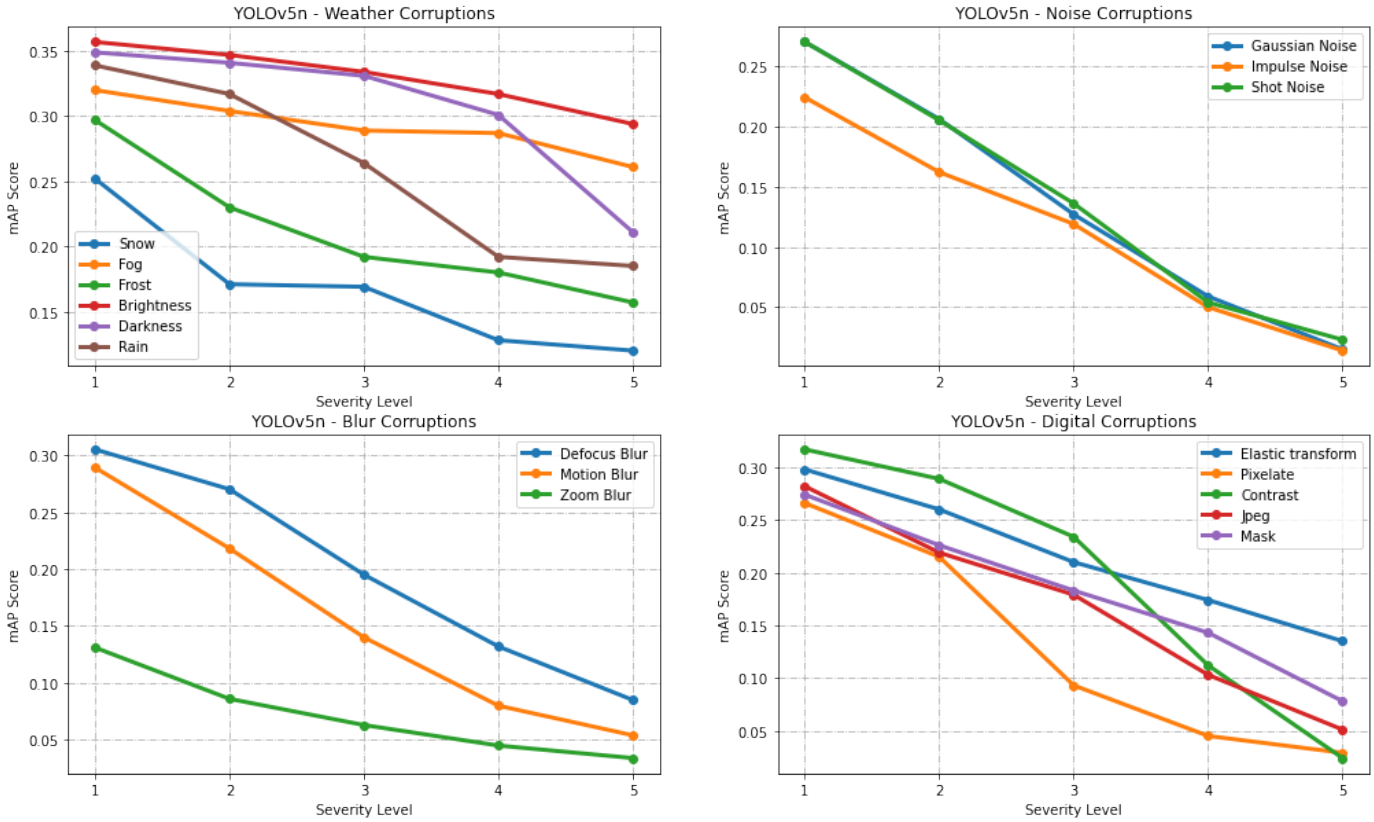


Figure 8.0.1: YOLOv5n performance on Corrupted COCO dataset

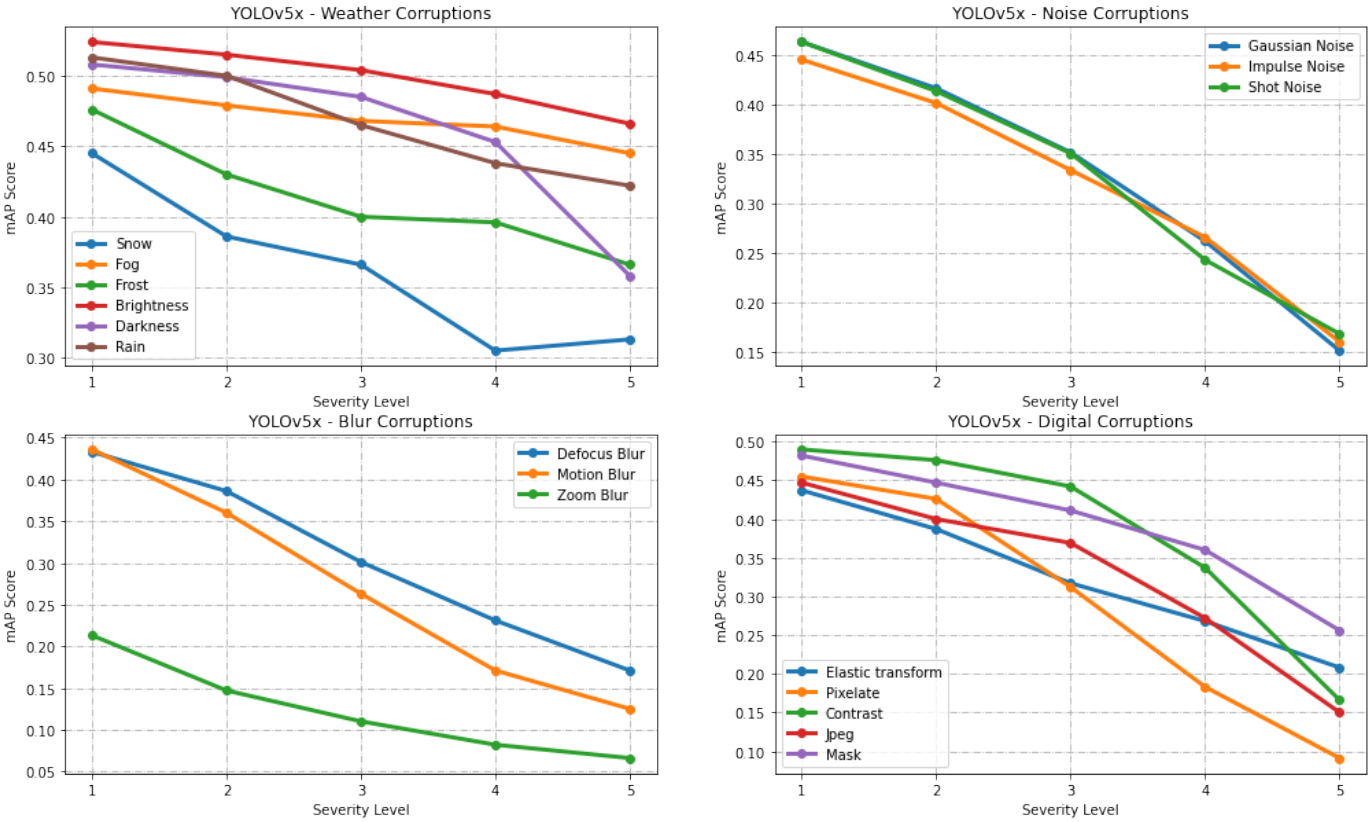


Figure 8.0.2: YOLOv5x performance on Corrupted COCO dataset

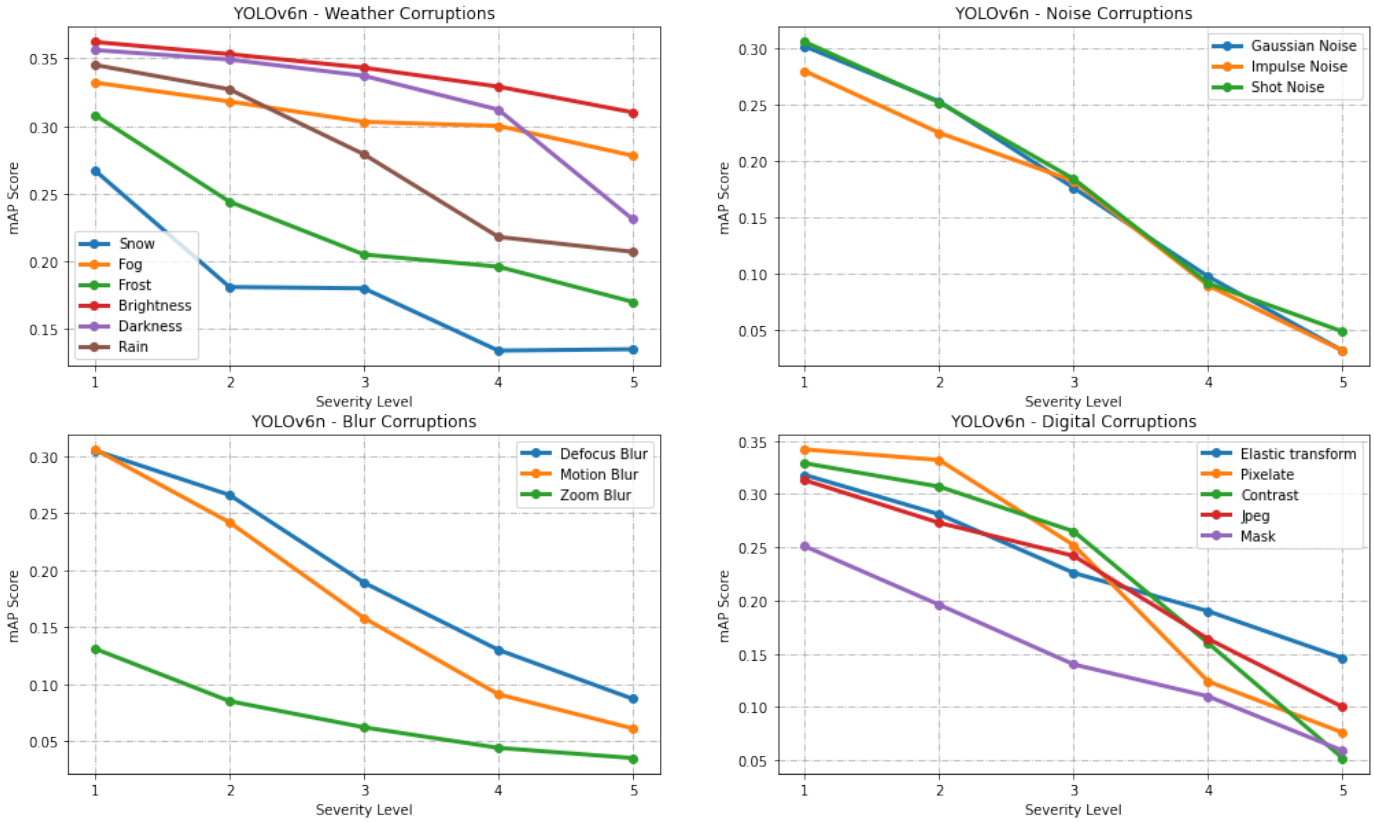


Figure 8.0.3: YOLOv6n performance on Corrupted COCO dataset

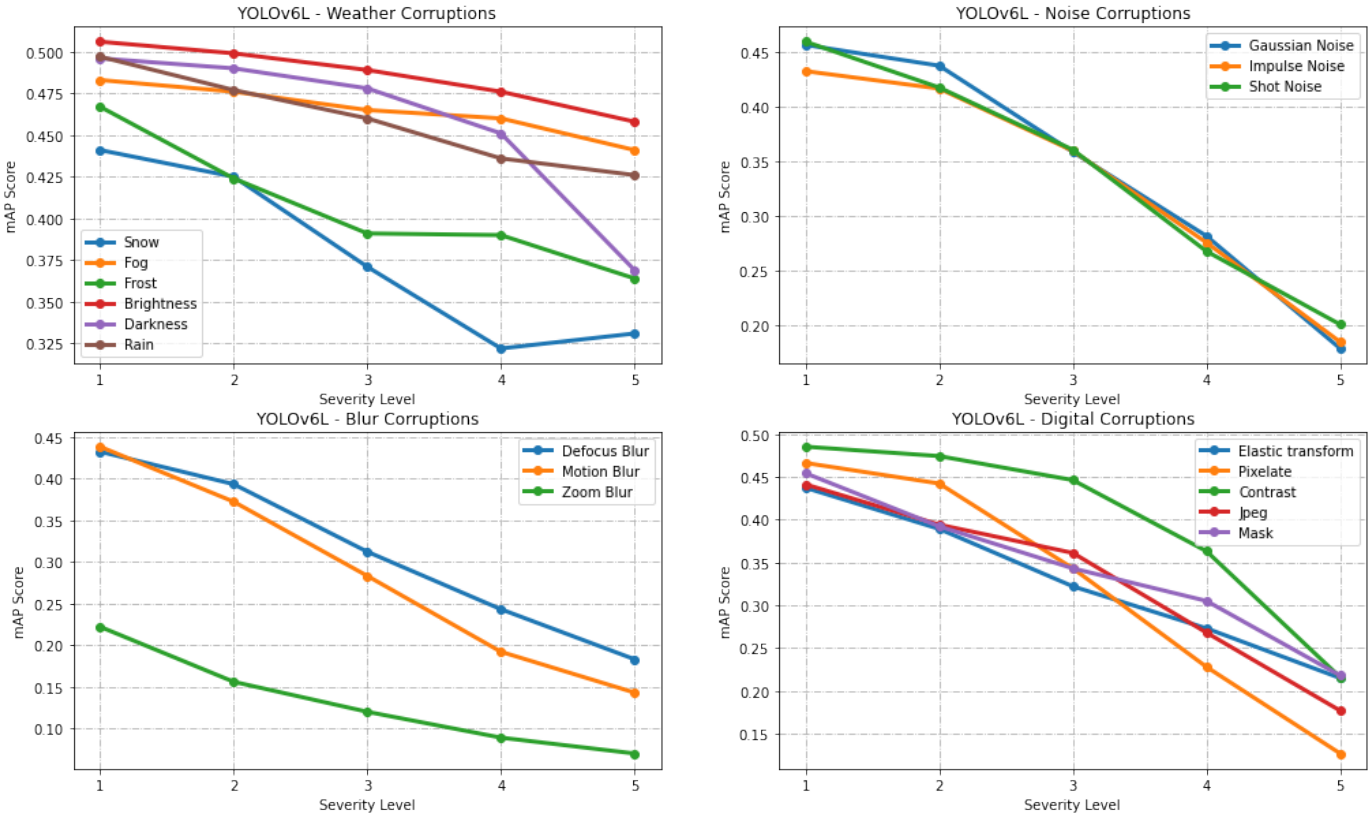


Figure 8.0.4: YOLOv6L performance on Corrupted COCO dataset

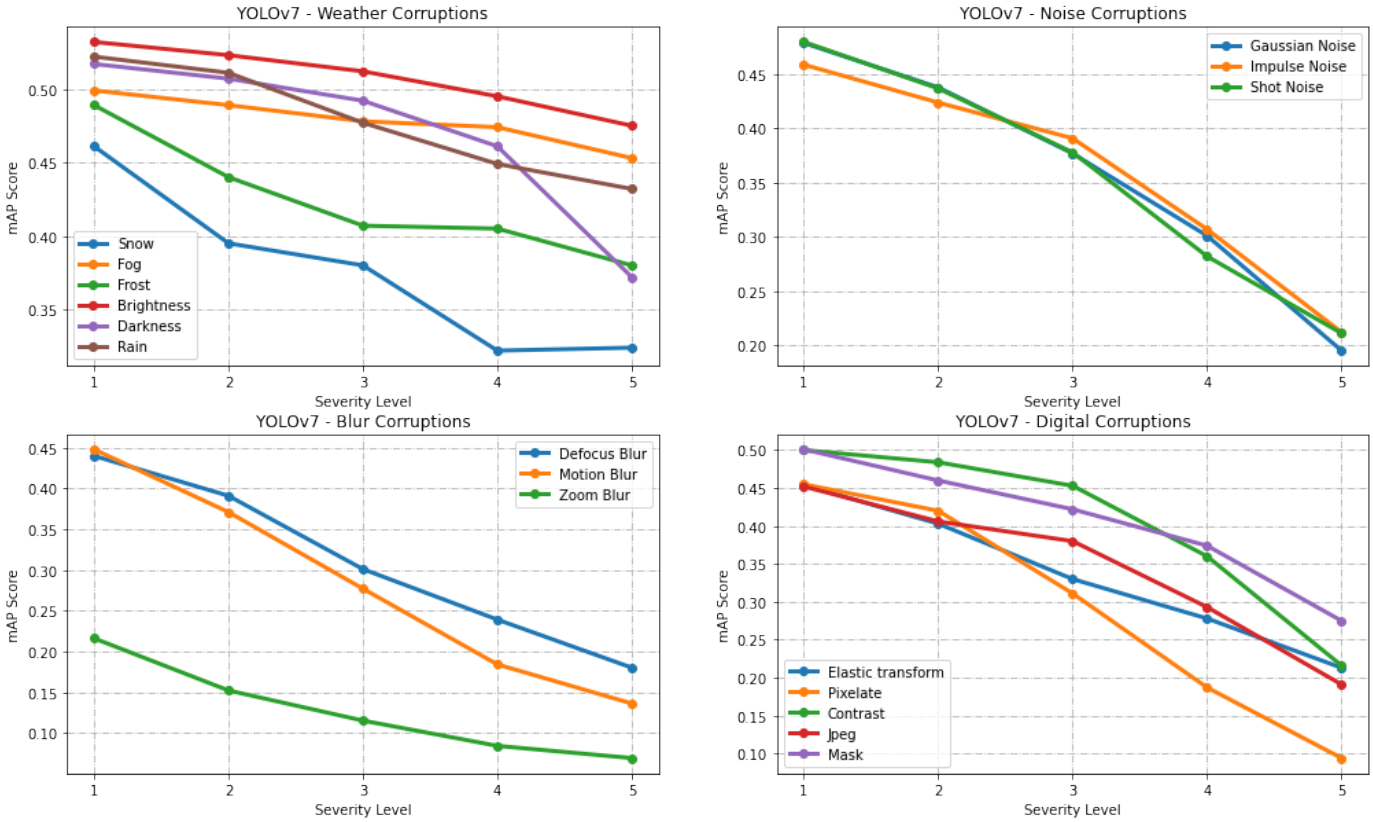


Figure 8.0.5: YOLOv7 performance on Corrupted COCO dataset

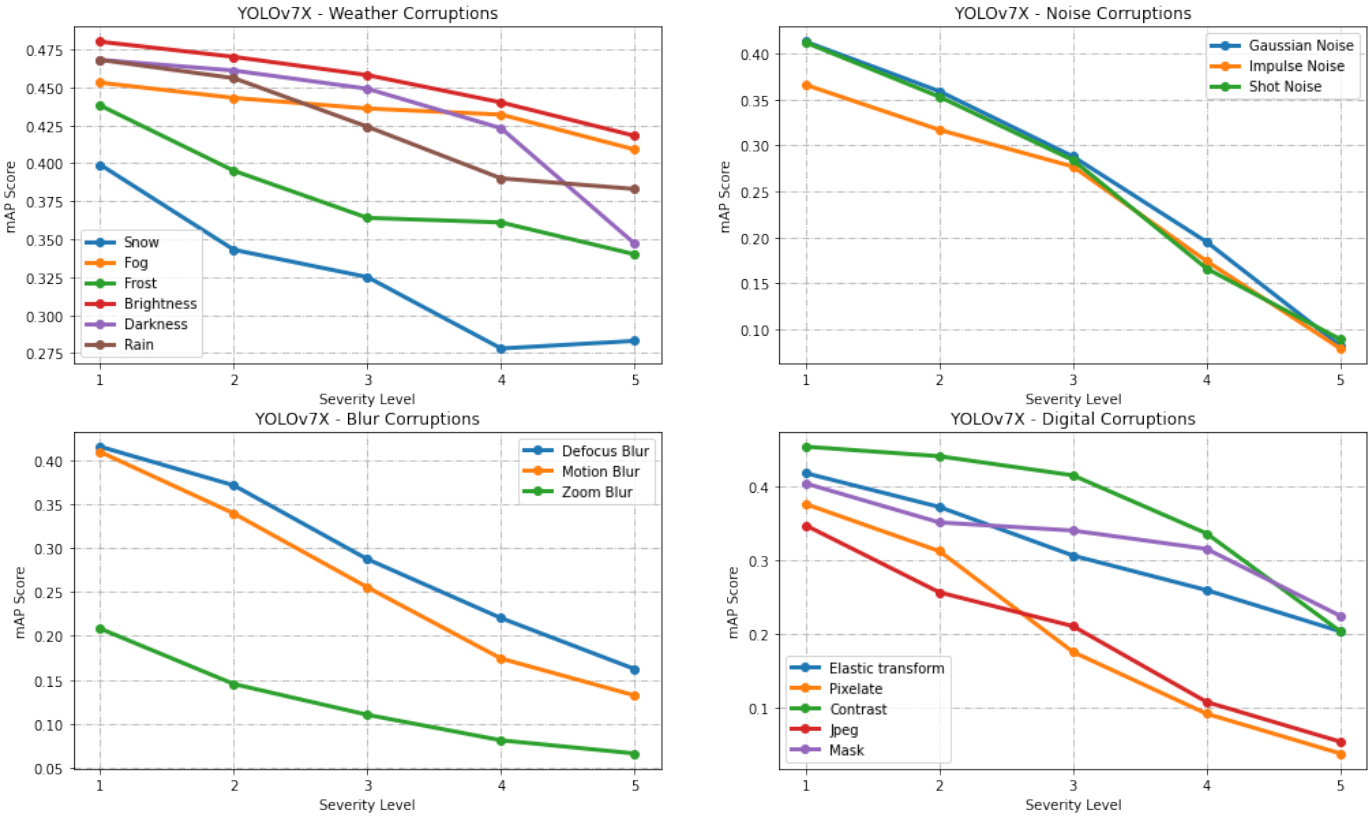


Figure 8.0.6: YOLOv7x performance on Corrupted COCO dataset

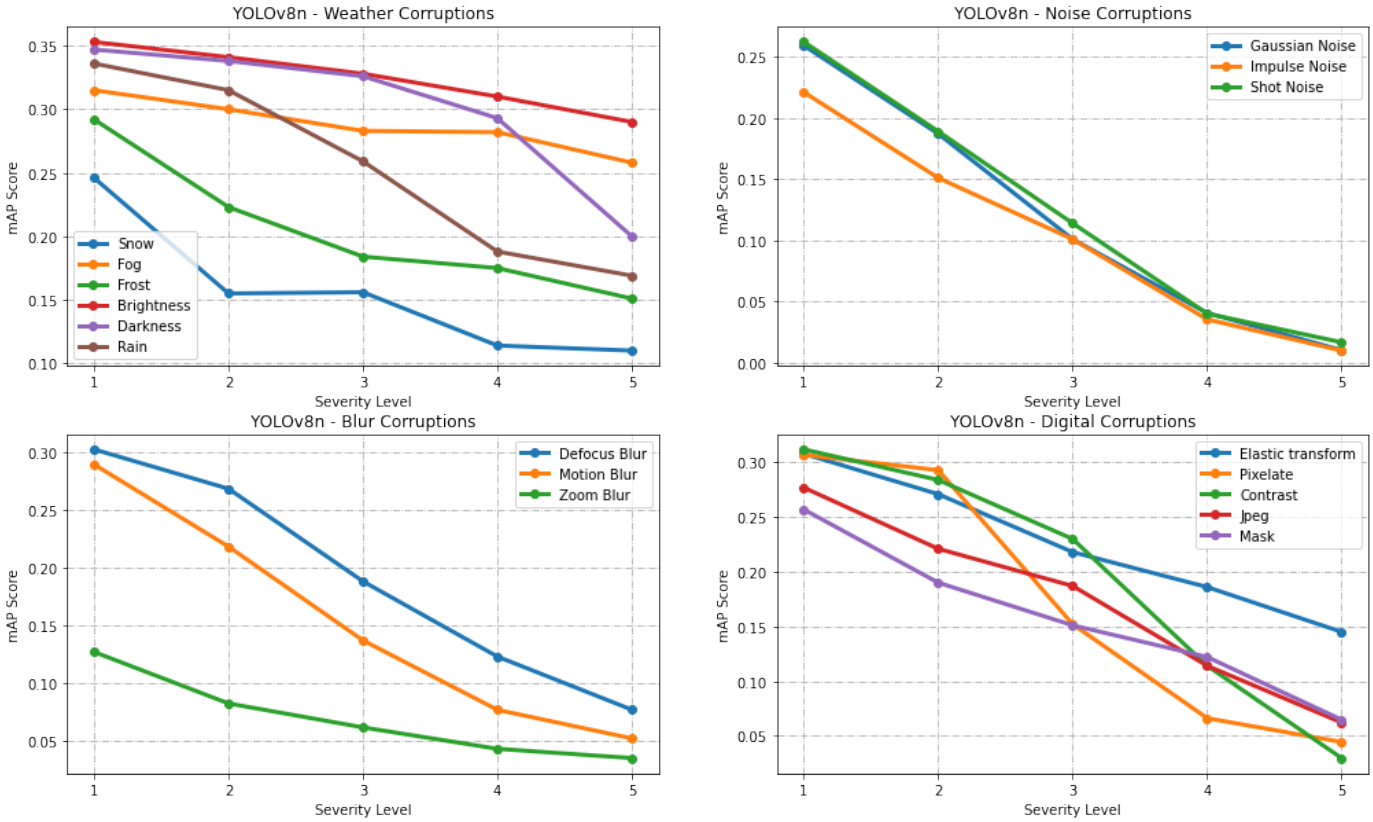


Figure 8.0.7: YOLOv8n performance on Corrupted COCO dataset

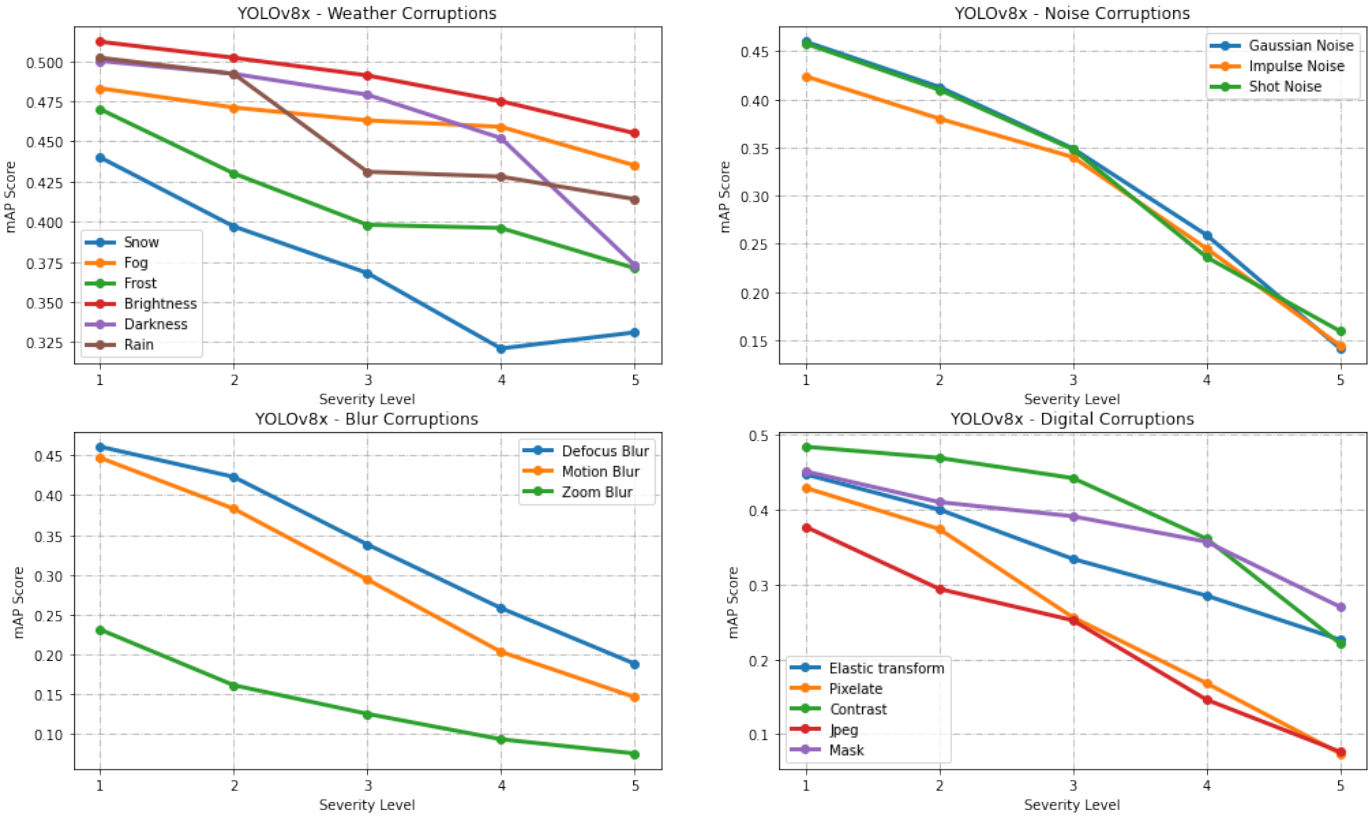


Figure 8.0.8: YOLOv8x performance on Corrupted COCO dataset

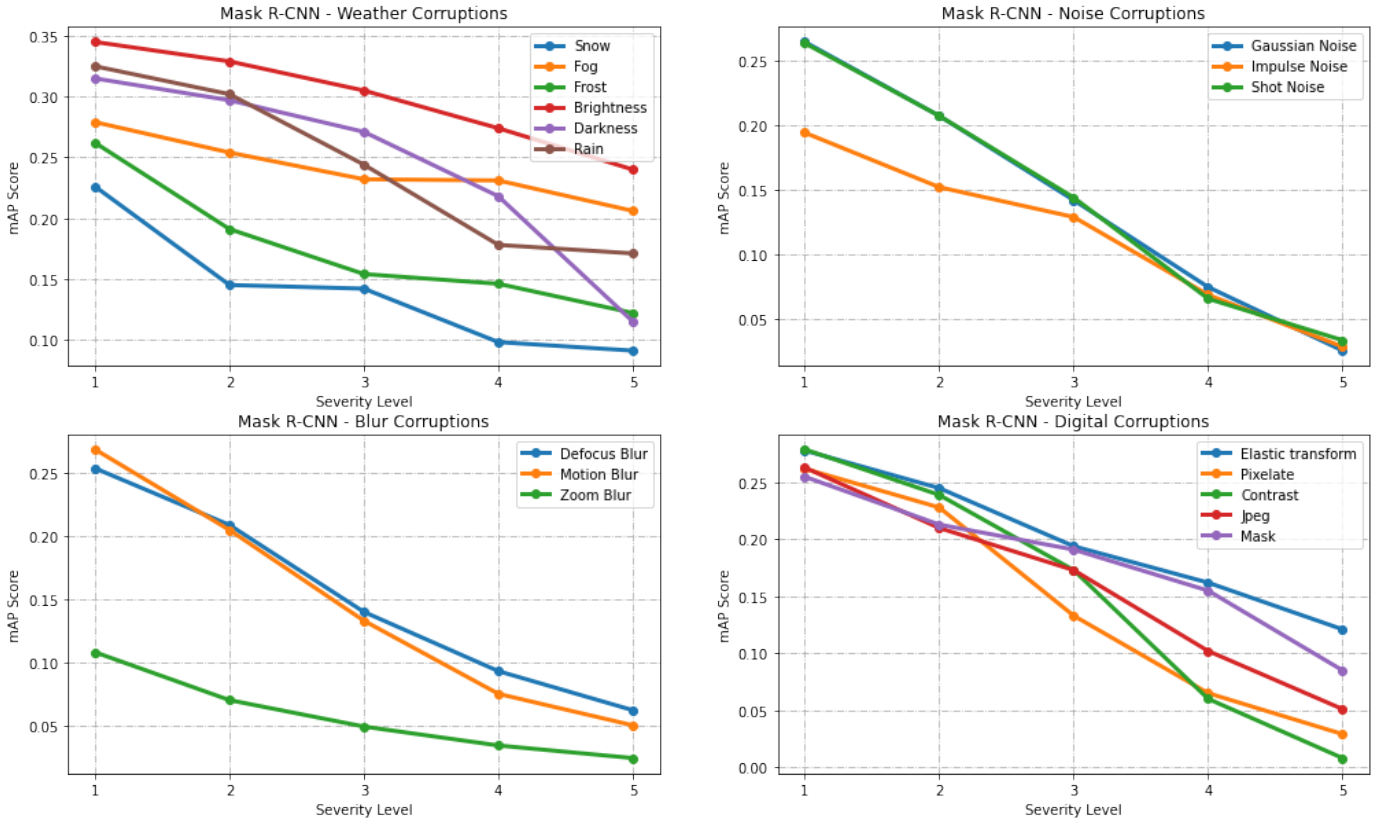


Figure 8.0.9: Mask R-CNN performance on Corrupted COCO dataset

Detector	YOLOv5n	YOLOv6n	YOLOv7	YOLOv8n	Mask RCNN
Snow	0.044	0.044	0.039	0.045	0.045
Frost	0.047	0.046	0.033	0.047	0.047
Fog	0.02	0.018	0.015	0.015	0.24
Brightness	0.019	0.017	0.021	0.021	0.035
Darken	0.046	0.042	0.04	0.049	0.067
Rain	0.051	0.046	0.028	0.056	0.051
Gauss	0.085	0.09	0.11	0.083	0.079
Impulse	0.07	0.083	0.096	0.07	0.055
Shot	0.082	0.086	0.108	0.082	0.076
Defocus	0.073	0.073	0.084	0.075	0.064
Zoom	0.032	0.032	0.047	0.031	0.028
Motion	0.078	0.082	0.104	0.079	0.073
Jpeg	0.077	0.071	0.087	0.072	0.071
Contrast	0.098	0.092	0.084	0.094	0.09
Pixelate	0.079	0.089	0.113	0.088	0.078
Elastic	0.054	0.057	0.072	0.054	0.052
Mask	0.065	0.064	0.06	0.064	0.057

Table 8.11: Absolute GmAP scores for all Small detectors and Corruptions

Detector	**YOLOv5x**	**YOLOv6l**	**YOLOv7E6**	**YOLOv8x**
Snow	0.044	0.037	0.046	0.036
Frost	0.037	0.034	0.036	0.033
Fog	0.015	0.014	0.015	0.016
Brightness	0.021	0.016	0.019	0.019
Darken	0.05	0.042	0.048	0.042
Rain	0.03	0.024	0.03	0.029
Gauss	0.104	0.093	0.095	0.106
Impulse	0.095	0.083	0.082	0.093
Shot	0.099	0.086	0.09	0.1
Defocus	0.087	0.083	0.087	0.091
Zoom	0.049	0.051	0.049	0.052
Motion	0.104	0.098	0.1048	0.1
Jpeg	0.099	0.088	0.087	0.1
Contrast	0.108	0.09	0.095	0.088
Pixelate	0.121	0.113	0.12	0.118
Elastic	0.076	0.074	0.08	0.074
Mask	0.075	0.079	0.075	0.06

Table 8.12: Absolute GmAP scores for all Large Detectors and Corruptions