Εθνικο Μετσοβιο Πολυτεχνειο

Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων

Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Εργαστηριο Συστηματων Τεχνητης Νοημοσυνης και Μαθησης

Camouflaged Object Detection and Segmentation

Diploma Thesis

by

Stratakis Michail

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

Camouflaged Object Detection and Segmentation

Diploma Thesis

by

Stratakis Michail

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)

.....

Στρατακης Μιχαηλ
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η ανίχνευση και η τμηματοποίηση καμουφλαρισμένων αντικειμένων αποτελεί ένα τμήμα της όρασης υπολογιστών που αποσκοπεί στην εύρεση αντικειμένων που δύσκολα ανιχνεύονται από το ανθρώπινο μάτι. Πρόκειται για μια διαδικασία αντίθετη από την εύρεση προεξέχοντος αντικείμενου (Salient Object Detection), όπου τα τμήματα της εικόνας προς ανίχνευση είναι διακριτά, εύκολα αναγνωρίσιμα και τα όρια τους διαφοροποιούνται αρκετά από το υπόλοιπο παρασκηνιακό περιβάλλον. Στην περίπτωση των καμουφλαρισμένων αντικειμένων, τα τμήματα της εικόνας προς ανίχνευση συχνά εμφανίζουν μεγάλη ομοιότητα με το υπόλοιπο περιβάλλον, αυξάνοντας αρκετά τη δυσκολία και τις προκλήσεις που πρέπει να αντιμετωπιστούν για τη διεκπεραίωση αυτού του έργου.

Στην παρούσα διπλωματική εργασία εισάγουμε μια νέα αρχιτεκτονική που συνδυάζει τις ισχυρές δυνατότητες των κωδικοποιητών από μετασχηματιστές (Transformer Encoder), για την εξαγωγή παγκόσμιων χαρακτηριστικών, με τις προϋπάρχουσες δομές των συνελικτικών κωδικοποιητών (Convolutional Encoder) για τη σύλληψη τοπικών χαρακτηριστικών. Κατά την ανίχνευση ενός καμουφλαρισμένου αντικειμένου είναι αρκετά δύσκολο να εντοπιστούν οι ακριβείς λεπτομέρειες κοντά στα άκρα του. Εμπνευσμένοι από αυτή την ικανότητα που διαθέτουν αυτά τα αντικείμενα, εισάγουμε ένα νέο μηχανισμό συνδυασμού των εξαγόμενων χαρακτηριστικών από τους δύο αυτούς κωδικοποιητές με σκοπό την παραγωγή πλούσιων γνωρισμάτων τόσο σε επίπεδο λεπτομέρειας όσο και σε επίπεδο σημασιολογικής ερμηνείας.

Αξιολογούμε και συγκρίνουμε το μοντέλο μας σε κοινά σύνολα δεδομένων και με κοινές μετρήσεις αξιολόγησης και παρουσιάζουμε τα ευρήματα μας. Τα ληφθέντα αποτελέσματα είναι αρκετά ενθαρρυντικά, καθώς πετυχαίνουν εξαιρετικές επιδόσεις και είναι ικανά να σταθούν αντάξια απέναντι στις τελευταίες βιβλιογραφικές τεχνολογίες. Τέλος, παρατηρούμε πως μεταβάλλεται η επίδοση του μοντέλου μας, καθώς τροποποιούμε είτε τον αλγόριθμο είτε κάποιες παραμέτρους και εν συνεχεία επισημαίνουμε πιθανές χρήσεις του μοντέλου μας για ιατρικούς και διάφορους άλλους σκοπούς.

Λέξεις-κλειδιά — Καμουφλαρισμένα Αντικείμενα, Τμηματοποίηση Εικόνας, Κωδικοποιητές Μετασχηματιστών, Συνελικτικοί Κωδικοποιητές, Συγχώνευση Χαρακτηριστικών, Αυτό-Προσοχή

Abstract

Camouflaged object detection and segmentation is a branch of computer vision that aims to find objects that are difficult to detect by the human eye. This is a process opposite to finding a salient object, where the parts of the image to be detected are distinct, easily recognizable and their boundaries are differentiated enough from the rest of the background environment. In the case of camouflaged objects, the parts of the image to be detected often show a high similarity to the rest of the environment, greatly increasing the difficulty and challenges that must be faced to carry out this task.

In this thesis we introduce a new architecture that combines the powerful capabilities of Transformer Encoders, for extracting global features, with the existing structures of Convolutional Encoders for capturing local features. When detecting a camouflaged object it is quite difficult to detect the exact details near its edges. Inspired by this ability of these objects, we introduce a novel method for combining the extracted features from these two encoders in order to produce rich features both at the level of detail and at the level of semantic interpretation.

We evaluate and compare our model on common datasets and with common evaluation metrics and present our findings. The results obtained are quite encouraging as they achieve excellent performance and are able to stand up against the latest technologies in literature. Finally, we observe how the performance of our model changes as we modify either the algorithm or some parameters and afterwards we point out possible uses of our model for medical and other purposes.

Keywords — Camouflaged Object, Image Segmentation, Transformer Encoder, Convolutional Encoder, Fuse Features, Self-Attention

Ευχαριστίες

Ολοκληρώνοντας τη διπλωματική και ταυτόχρονα τις σπουδές μου στο Εθνικό Μετσόβειο Πολυτεχνείο, οφείλω να εκφράσω την τεράστια ευγνωμοσύνη μου προς εκείνους τους ανθρώπους που συνετέλεσαν θετικά σε αυτό το αποτέλεσμα. Ευχαριστώ θερμά τον επιβλέπων καθηγητή μου, κύριο Στάμου, για τις πολύτιμες υποδείξεις του, την επιμονή και το αμείωτο ενδιαφέρον του στον ευρύτερο τομέα της τεχνητής νοημοσύνης. Επίσης, ευχαριστώ τον καθηγητή, κύριο Αλεξανδρίδη, για την εμπιστοσύνη που μου έδειξε εξαρχής ώστε να αναλάβω το συγκεκριμένο θέμα μετά από αρκετές εποικοδομητικές συζητήσεις. Θα ήθελα ακόμα να ευχαριστήσω τη διδακτορικό Μαρία Λυμπεραίου για τις συμβουλές, τις γνώσεις και τη θερμή της καθοδήγηση, καθώς η ανταπόκρισή της στις ανάγκες που προέκυπταν τόσο κατά την έρευνα όσο και κατά τη συγγραφή ήταν πάντα άμεση.

Τέλος, αισθάνομαι βαθύτατα την ανάγκη να ευχαριστήσω την οικογένειά μου, καθώς υπήρξε ένα τεράστιο ψυχολογικό στήριγμα και ταυτόχρονα έδειξε κατανόηση στο χρόνο που αφιέρωσα για την ολοκλήρωση των σπουδών με παράλληλη εργασία.

Στρατάκης Μιχαήλ,
Αθήνα, Μάρτιος 2023

Contents

List of Figures

List of Tables

Εκτεταμένη Περίληψη

Θεωρητικό Υπόβαθρο

Η όραση υπολογιστών είναι ένα πεδίο τεχνητής νοημοσύνης που επεξεργάζεται οπτικά δεδομένα χρησιμοποιώντας σε μεγάλο βαθμό τη Μηχανική Μάθηση μέσω τεχνικών Βαθιάς Μάθησης. Συγκεκριμένα, την τελευταία δεκαετία, τα βαθιά συνελικτικά δίκτυα έχουν δημιουργήσει πολλά επιτεύγματα στις διάφορες πτυχές της όρασης υπολογιστών. Μέσα από αυτές τις καινοτομίες είναι δυνατόν για ένα μοντέλο τεχνητής νοημοσύνης να αναγνωρίζει αντικείμενα και μοτίβα όπως το ανθρώπινο μάτι και να κατανοεί σημασιολογικά τον κόσμο, όπως ο ανθρώπινος νους.

Οι σύγχρονες εφαρμογές υπολογιστικής όρασης, παρά το γεγονός ότι έχουν βαριές αρχιτεκτονικές παραμέτρων, έχουν παρουσιάσει μεγάλες επιτυχίες μέσα από την αξιοποίηση μονάδων γραφικής επεξεργασίας (GPUs) κατά τη διαδικασία της εκπαίδευσης, με αποτέλεσμα ταχύτερη επεξεργασία δεδομένων και συστημάτων ικανών να επιτύχουν υψηλή ακρίβεια και όσο το δυνατόν λιγότερες ψευδείς προβλέψεις [39]. Σημαντικές υπηρεσιές που εκτελούνται σε καθημερινή βάση συνήθως χρησιμοποιούν βασικές έννοιες και τεχνικές της όρασης υπολογιστών. Με αυτό τον τρόπο, αποδεικνύεται ο σημαντικός ρόλος της όρασης υπολογιστών στην ανίχνευση αντικειμένων και τη σημασιολογική τμηματοποίηση στην υγειονομική περίθαλψη, καθώς και σε διάφορους τομείς παραγωγής.

Η Όραση Υπολογιστών ως πεδίο έρευνας ενέχει αρκετές προκλήσεις, κυρίως λόγω της άψογης απόδοσης του ανθρώπινου ματιού σε αναρίθμητες οπτικές προκλήσεις [22]. Για να αποκτήσει μία αντίστοιχη απόδοση ένα τεχνητό μοντέλο, πρέπει να διενεργηθεί ένας μεγάλος αριθμός υπολογισμών και να εξασφαλιστεί η ύπαρξη αρκετών οπτικών δεδομένων. Για περαιτέρω βελτίωση του αποτελέσματος και ελαχιστοποίηση της υπερπροσαρμογής τα εκάστοτε μοντέλα απαιτούν αρχιτεκτονικές, με σκοπό την απόκτηση ουσιαστικής πληροφορίας και την κατανόηση των παρεχόμενων δεδομένων σε επίπεδο σημασιολογικό.

Συμβολή και σκοπός του προτεινόμενου νευρωνικού δικτύου

Μία από τις βασικές ικανότητες του ανθρώπινου ματιού βασίζεται στην επιτυχή και στιγμιαία αναγνώριση αντικειμένων και στη σημασιολογική κατανόηση του κόσμου. Υπό πολλές προϋποθέσεις η αναγνώριση των αντικειμένων και η κατανόησή τους είναι ακριβής και η διαδικασία έρχεται φυσικά και αβίαστα. Αντίθετα, τα μοντέλα υπολογιστικής όρασης απαιτούν την εξαγωγή ενός συνόλου χαρακτηριστικών από την ψηφιακή αναπαράσταση του κόσμου και ύστερα την ανάλυσή τους.

Υπάρχουν αντικείμενα είτε φυσικά είτε τεχνητά κατασκευασμένα με δυνατότητα καμουφλάζ, δηλαδή αποφεύγουν να προδώσουν τα σημαντικά και προεξέχοντα χαρακτηριστικά τους. Η εξέλιξη των ζώων στην άγρια φύση και το ανθρωπογενές καμουφλάζ μειώνουν την πιθανότητα ανίχνευσης ή αναγνώρισης από πιθανούς θηρευτές ή εχθρούς [55]. Επιπλέον, η εξέλιξη των νέων ασθενειών δημιουργεί ένα εμπόδιο στη σύγχρονη βιομηχανία υγειονομικής περίθαλψης όταν πρόκειται για ανίχνευση πνευμονικών λοιμώξεων από ιατρικές εικόνες [45]. Μια ευρέως χρησιμοποιούμενη στρατηγική από τέτοια αντικείμενα είναι να προσαρμοστούν στο σχέδιο, το χρώμα και άλλες μορφολογικές ιδιότητες του περιβάλλοντος σε τέτοιο βαθμό, ώστε η ανίχνευσή τους να οδηγήσει ακόμη και το έμπειρο ανθρώπινο μάτι σε ψευδή απόφαση [43]. Συνολικά, το καμουφλάζ χειραγωγεί την οπτική αναπαράσταση που φτάνει στον θεατή και αυξάνει σημαντικά τις προκλήσεις μιας ακριβούς τμηματοποίησης [37].

Με την εισαγωγή της αρχιτεκτονικής των Transformers επήλθε μια επανάσταση στον τρόπο με τον οποίο σύγχρονα μοντέλα μάθησης επικεντρώνονται και δίνουν προσοχή στα σημαντικά στοιχεία μιας εικόνας, καθώς

δημιουργούν εξαρτήσεις σε επίπεδο καθολικό [60]. Από την άλλη μεριά, τα παραδοσιακά συνελικτικά δίκτυα (CNNs: Convolutional Neural Networks) εξάγουν χαρακτηριστικά σαρώνοντας την εκάστοτε εικόνα με βάση το μέγεθος του πυρήνα (kernel) [40]. Η διαδικασία αυτή στα ρηχά επίπεδα ενσωματώνει πληροφορίες χαμηλού επιπέδου, όπως οι ακμές και το σχήμα των αντικειμένων, ενώ σε βαθύτερα επίπεδα η πληροφορία εμπεριέχει τη σημασιολογική αναπαράσταση του κόσμου [19, 28].

Σκοπός αυτής της διπλωματικής είναι να θέσει τα θεμέλια για τη δημιουργία μίας επιτυχούς σύζευξης των CNNs με τη νέα τεχνολογία των Transformers. Γίνεται προσπάθεια για κατάλληλη σύνδεση των χαρακτηριστικών χαμηλού επιπέδου από τα παραδοσιακά συνελικτικά νευρωνικά δίκτυα με την καθολική πληροφορία από τους Transformers. Πιο συγκεκριμένα, η ραχοκοκαλιά του μοντέλου μας με την ονομασία **RACOD-Net** αποτελείται από δύο κωδικοποιητές, τον ResNet50 [18] που αντιπροσωπεύει τη γενιά των συνελικτικών νευρωνικών δικτύων με τον SegFormer [68] που υλοποιεί την τεχνολογία των Tranformers. Στη συνέχεια, ο μερικός κλιμακωτός αποκωδικοποιητής μας κατασκευάζει τη σειρά με την οποία συνδέονται τα εξαγόμενα χαρακτηριστικά που παράχθηκαν νωρίτερα από τους κωδικοποιητές. Όπως φαίνεται στο Σχ. 1, το μοντέλο μας παρουσιάζει εξαιρετικά οπτικά αποτελέσματα και ξεπερνάει αρκετές προηγούμενες έρευνες στις παραγόμενες προβλέψεις.

Σχετικές Προηγούμενες Εργασίες

Όπως υποδηλώνει ο όρος, η κατάτμηση εικόνας είναι η διαδικασία διχοτόμησης μιας εικόνας σε πολλαπλά τμήματα. Υπάρχουν δύο κύριοι τύποι τμηματοποίησης εικόνας, η σημασιολογική τμηματοποίηση (semantic segmentation) όταν όλα τα αντικείμενα που ανήκουν στον ίδιο τύπο χαρακτηρίζονται από την ίδια ετικέτα κλάσης και η τμηματοποίηση ανά περίσταση (instance segmentation) όπου όλες οι εμφανίσεις του ίδιου τύπου ταυτοποιούνται με διαφορετική ετικέτα. Η ανίχνευση και η τμηματοποίηση καμουφλαρισμένων αντικειμένων ανήκει προς το παρόν στην κατηγορία της σημασιολογικής τμηματοποίησης. Κατά συνέπεια, στα εικονοστοιχεία (pixels) που ανήκουν σε κάποιο καμουφλαρισμένο αντικείμενο θα ανατίθεται η τιμή 1, ενώ στην αντίθετη περίπτωση θα ανατίθεται η τιμή 0. Αυτό καθιστά το συγκεκριμένο αντικείμενο της όρασης υπολογιστών και ως μία δυαδική τμηματοποίηση εικόνας.

Αναρίθμητες προηγούμενες έρευνες έχουν διεξαχθεί χρησιμοποιώντας κατά κύριο λόγο CNNs και έχουν παρουσιάσει εξαιρετικά αποτελέσμα. Ανάμεσα σε αυτές τις έρευνες συγκαταλέγονται σπουδαία μοντέλα όπως το BASNet [44], το SINet [11], το ANet [27], το SINet-V2 [9], το BSANet [79], το BGNet [58], καθώς και το PFNet [36] τα οποία κάνουν χρήση ενός συνελικτικού νευρωνικού κωδικοποιητή και ύστερα μέσω κατάλληλου αποκωδικοποιητή παράγουν αρχικά μία πρώιμη πρόβλεψη την οποία μέσα από τεχνικές βελτιστοποίησης ενισχύουν με τελικό αποτέλεσμα μία πιο ακριβή πρόβλεψη.

Παράλληλα με την εξέλιξη των Transformers, το συγκεκριμένο αντικείμενο βελτιώθηκε σε μεγάλο βαθμό. Οι μελέτες, κάνοντας χρήση αυτής της τεχνολογίας, είναι λιγότερες συγκριτικά με τις προηγούμενες. Ενδεικτικές αποτελούν οι αρχιτεκτονικές του DTINet [31], του UGTR [69] και του CamoFormer [70] που ενσωματώνουν στους κωδικοποιητές τους στοιχεία από τους Transformers.

Σχήμα 1: Οπτικές συγκρίσεις ανάμεσα σε καμουφλαρισμένα αντικείμενα, μεταξύ του δικού μας μοντέλου RACOD-Net και των υπόλοιπων μοντέλων της διεθνούς βιβλιογραφίας.

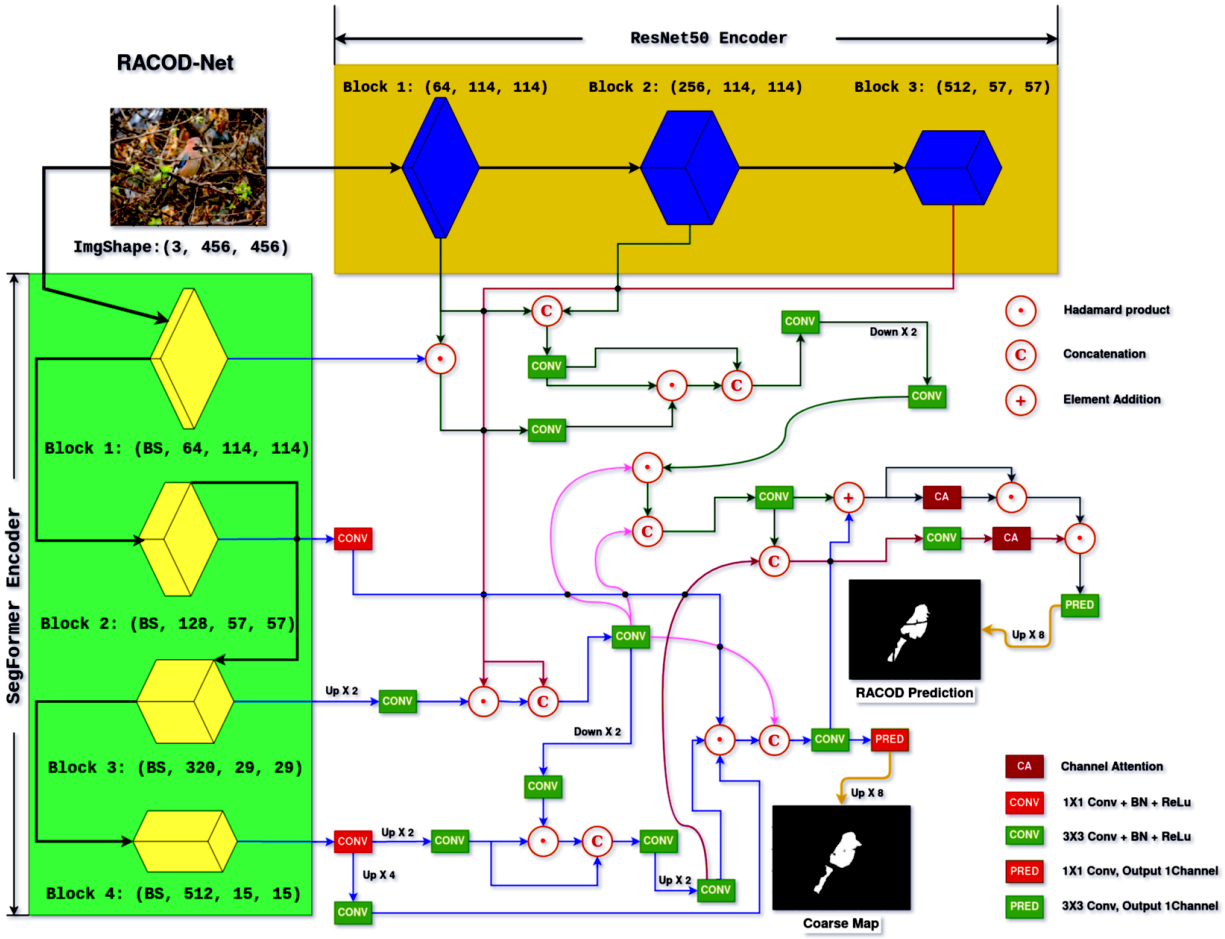
Η αρχιτεκτονική του RACOD-Net

Άμεσος στόχος της αρχιτεκτονικής μας είναι ο κατάλληλος συνδυασμός δύο κωδικοποιητών που υλοποιούν διαφορετικές φιλοσοφίες. Μέσα από έναν πρωτοποριακό κλιμακωτό αποκωδικοποιητή ενοποιούνται τα εξαγόμενα χαρακτηριστικά από τον Transformer κωδικοποιητή και τον αντίστοιχο συνελικτικό νευρωνικό κωδικοποιητή.

Κατά συνέπεια, παράγεται ένα τελικό αποτέλεσμα με πλούσια πληροφορία τόσο τοπική όσο και καθολική. Όπως απεικονίζεται στο Σχ. 2, αρχικά εξάγονται στο σύνολο έφτα χαρακτηριστικά από τους κωδικοποιητές μας. Στο σύνολο $\{X_k\}_{k=1}^{3}$ χαρακτηριστικά από τον ResNet50 [18] και $\{C_k\}_{k=1}^{4}$ χαρακτηριστικά από τον SegFormer [68].

Ο CNN κωδικοποιητής

Η αρχιτεκτονική ResNet50 καταφέρνει να επιλύσει το πρόβλημα των εξαφανιζόμενων κλίσεων. Κατά το πέρασμα προς τα πίσω, τα βάρη των στρωμάτων κοντά στην είσοδο παραμένουν σταθερά ή ενημερώνονται πολύ αργά, σε αντίθεση με αυτό που συμβαίνει στα επίπεδα κοντά στην έξοδο. Αυτό το πρόβλημα οδηγεί σε κορεσμό των επιδόσεων από ένα βάθος και μετά και επιλύεται από τον ResNet50 με την εισαγωγή συνδέσεων που αθροίζουν τις εξόδους από τα ρηχότερα επίπεδα στις εξόδους των επόμενων βαθύτερων επιπέδων. Στη δική μας περίπτωση, χρησιμοποιούμε αυτή την αρχιτεκτονική για να συλλέξουμε τα πρώτα τρία ρηχά επίπεδα.

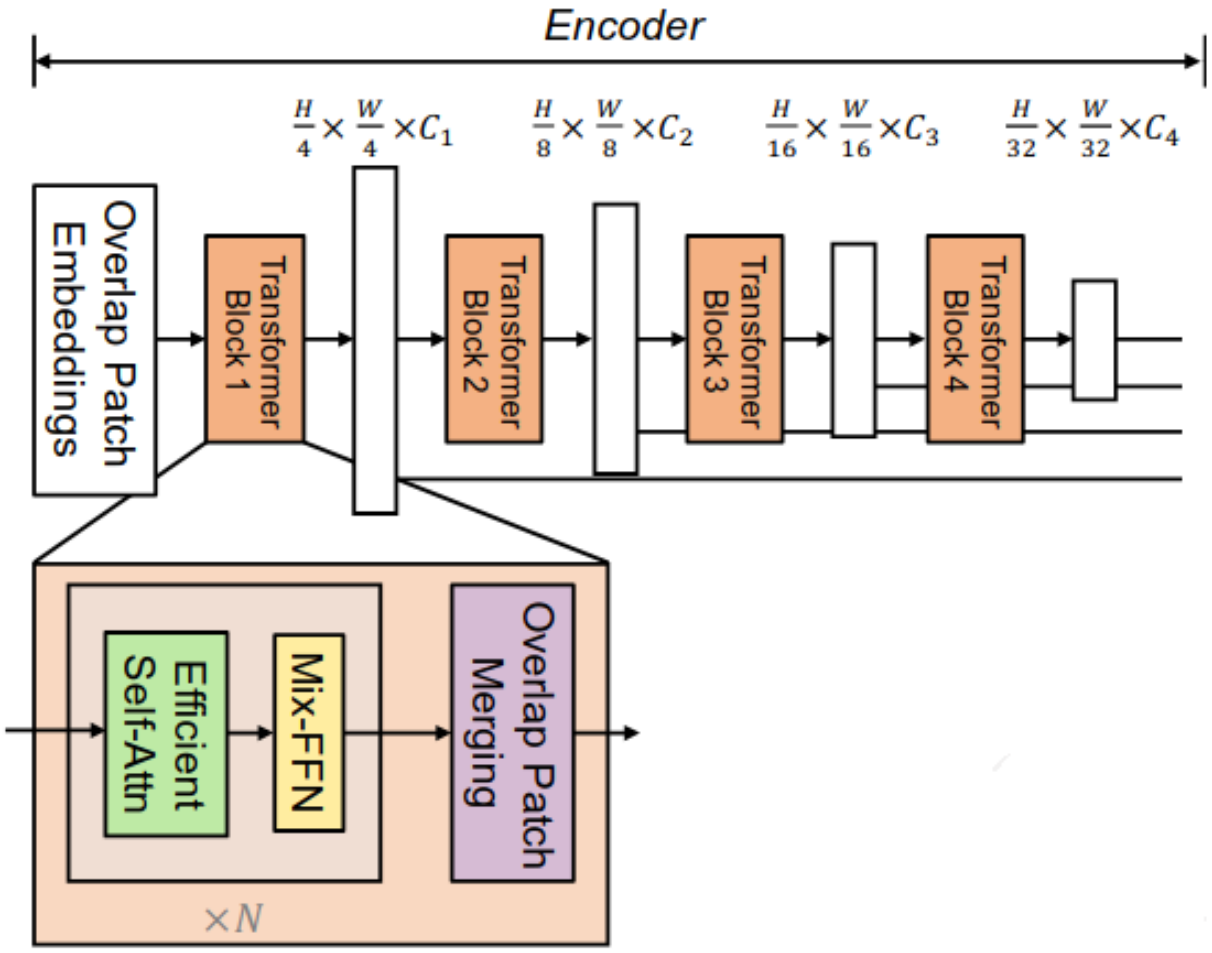


Σχήμα 2: Η αρχιτεκτονική του RACOD-Net

Ο Transformer κωδικοποιητής

Η βασική ιδέα πίσω από τον SegFormer οπτικοποιείται στο Σχ. 3. Τα τέσσερα εξαγόμενα χαρακτηριστικά από τον SegFormer ακολουθούν την παραδοσιακή ιεραρχία της Βαθιάς Μάθησης, όπου καθώς προχωράμε σε βαθύτερα επίπεδα οι χωρικές διαστάσεις μειώνονται. Το πρώτο βήμα προς αυτή την κατεύθυνση ξεκινάει με ένα στάδιο επικαλυπτόμενης διάσπασης της εικόνας σε μικρότερα κομμάτια. Αυτό επιτυγχάνεται με τη χρήση ενός πυρήνα (kernel) με μέγεθος (7, 7), βήμα (stride) 4 και γέμισμα (padding) 3 για το πρώτο επίπεδο, ενώ τα επόμενα επίπεδα αναμένουν μέγεθος πυρήνα (3, 3), βήμα 2 και γέμισμα 1. Με αυτόν τον τρόπο, τα κομμάτια στα οποία διασπάται η εικόνα έχουν επικαλυπτόμενα μέρη. Στη συνέχεια, το εκάστοτε επίπεδο επικαλυπτόμενης διάσπασης γίνεται είσοδος στο τμήμα που υλοποιεί τον μηχανισμό της αυτό-προσοχής (self-attention) όπως

αυτή εισήχθη στην έρευνα [60]. Μια καινοτομία του SegFormer είναι η εισαγωγή ενός συνελικτικού επιπέδου με μέγεθος πυρήνα (R, R) και βήμα R πριν τον υπολογισμό της αυτο-προσοχής, που περιλαμβάνει πολλαπλασιασμό πινάκων, μειώνοντας την πολυπλοκότητα από $\mathcal{O}(N^2)$ σε $\mathcal{O}\left(\frac{N^2}{R}\right)$, όπου $N = H \times W$ για μία εικόνα με διαστάσεις H×W . Για την ολοκλήρωση ενός επιπέδου η έξοδος που προκύπτει από την αυτο-προσοχή γίνεται είσοδος σε ένα δίκτυο ενιαίας τροφοδότησης μιας κατεύθυνσης (feed-forward network). Δηλαδή, ένα δίκτυο στο οποίο δεν υπάρχουν αναδρομές και η εξαγωγή του τελικού χαρακτηριστικού διέρχεται από πολλαπλά στάδια επεξεργασίας που περιλαμβάνουν γραμμικούς μετασχηματισμούς, συνελίξεις και συναρτήσεις ενεργοποίησης.



Σχήμα 3: Ο Transformer κωδικοποιητής [68]

Ο αποκωδικοποιητής του RACOD-Net

Ο αποκωδικοποιητής μας, βασισμένος στην αρχική ιδέα από την έρευνα [67], υλοποιεί μία τεχνική όπου παράγει δύο αποτελέσματα με κλιμακωτό τρόπο. Πρόκειται για μία τεχνική με δύο παραγόμενους κλάδους, η οποία έχει υιοθετηθεί από αναρίθμητα μοντέλα, όπως τα [13], [44], [11], [9], [29], [31], [5] και [70]. Στόχος είναι με κατάλληλο τρόπο να παράξουμε μία πρώιμη πρόβλεψη από τον πρώτο κλάδο, η οποία μαζί με κάποια σημεία-κλειδιά του ίδιου κλάδου ενοποιείται μέσω συγκεκριμένης μεθοδολογίας με χαρακτηριστικά του δεύτερου κλάδου. Το τελικό αποτέλεσμα του δεύτερου κλάδου οδηγεί τελικά σε μία πρόβλεψη που ενέχει μεγαλύτερη ακρίβεια και ευκρίνεια. Ο αποκωδικοποιητής βασίζεται στους κωδικοποιητές ResNet50 και SegFormer, που παράγουν τα $\{\mathrm{X}_k\}_{k=1}^{3}$ και τα $\{\mathrm{C}_k\}_{k=1}^{4}$ χαρακτηριστικά αντίστοιχα.

Η Πρώιμη Πρόβλεψη

Όπως γίνεται εύκολα αντιληπτό από το Σχ. 2, η πρώιμη πρόβλεψη (coarse map) συγχωνεύει κατάλληλα το τρίτο στοιχείο από τον ResNet50 και τα τρία τελευταία στοιχεία από τον SegFormer. Αυτή η αρχική πρόβλεψη, κατά συνέπεια, αποτελείται από χαρακτηριστικά, τα οποία συνδυάστηκαν ώστε το παραγόμενο αποτέλεσμα να ενέχει χαρακτήρα σημασιολογικό και καθολικό. Δηλαδή να είναι σε θέση να αντιληφθεί τη φύση και τη σημασιολογία του αντικειμένου στον χώρο. Από την ένωση των στοιχείων X_2 και C_3 δημιουργείται το ενδιάμεσο προϊόν με την ονομασία MF(Mixed Features), όπως απεικονίζεται παρακάτω:

$$MF = BConv(Concat(BConv(Inter(C_3)) \odot X_2),\, X_2), \tag{1}$$

όπου το επίπεδο BConv$(\cdot)$ περιλαμβάνει ένα στρώμα συνέλιξης, που ακολουθείται από ένα στρώμα κανονικοποίησης και τέλος από τη συνάρτηση ενεργοποίησης ReLU. Το επίπεδο Concat$(\cdot)$ αντιπροσωπεύει τη συγχώνευση χαρακτηριστικών στη διάσταση των καναλιών, το επίπεδο Inter$(\cdot)$ είναι υπεύθυνο για την αναδειγματολειψία του εκάστοτε χαρακτηριστικού ώστε να ταιριάξουν οι χωρικές του διαστάσεις με κάποιο άλλο χαρακτηριστικό ενώ το σύμβολο $\odot$ υποδευκνύει πολλαπλασιασμό ανάμεσα σε τανυστές που παράγονται κατά τη διάρκεια εκπαίδευσης του νευρωνικού μας δικτύου.

Ο σκοπός πίσω από ένα πολλαπλασιασμό στοιχείο προς στοιχείο, όπως αναφερέται στην έρευνα [11], αποσκοπεί στη μείωση του σημασιολογικού χάσματος ανάμεσα στα χαρακτηριστικά μας. Η αρχιτεκτονική του RACOD-Net Thus παρουσιάζει συχνά τμήματα που περιλαμβάνουν πολλαπλασιασμό, συγχώνευση και στη συνέχεια ένα επίπεδο BConv$(\cdot)$ με σκοπό τη συγκέντρωση μίας ενισχυμένης πληροφορίας σε ένα καινούργιο υβριδικό χαρακτηριστικό.

Στη συνέχεια, ο αποκωδικοποιητής μας αναμιγνύει τα στοιχεία C_3, C_4 και MF και εξάγεται το χαρακτηριστικό C_4C_3, όπως φαίνεται παρακάτω:

$$C_{4UpX2} = BConv(Inter(Bconv(C_4))) \tag{2}$$

$$C_3' = BConv(Inter(MF)) \tag{3}$$

$$C_4C_3 = BConv(Inter(BConv(Concat(C_{4UpX2} \odot C_3',\, C_{4UpX2})))), \tag{4}$$

Η πρώιμη πρόβλεψη, δηλαδή το τελικό αποτέλεσμα του πρώτου κλάδου του αποκωδικοποιητή μας γεννάται μέτα από τις παρακάτω σειριακές ενέργειες:

$$C_{4UpX4} = BConv(Inter(Bconv(C_4))) \tag{5}$$

$$C_2' = BConv(C_2)) \tag{6}$$

$$C_4C_3C_2 = C_2' \odot C_{4UpX4} \odot C_4C_3 \tag{7}$$

$$C_4C_3C_2MF = Bconv(Concat(C_4C_3C_2,\, MF)) \tag{8}$$

$$Coarse\ Map = Inter(Pred(C_4C_3C_2MF)), \tag{9}$$

όπου το στρώμα Pred$(\cdot)$ προσθέτει ένα ακόμα επιπεδο συνέλιξης με σκοπό τη μετάβαση της εξόδου μας σε ένα κανάλι (ασπρόμαυρη εικόνα). Ο πρώτος κλάδος ολοκληρώνεται με την εξίσωση (9), όπου η πρώιμη πρόβλεψη δειγματοληπτείται προς τα άνω οχτώ φορές προκειμένου να αποκτήσει τις ίδιες χωρικές διαστάσεις με την επαληθευμένη αλήθεια.

Η Τελική Ακριβής Πρόβλεψη

Όπως αναφέρεται στην έρευνα [66], τα βαθύτερα χαρακτηριστικά από τους Transformers περιλαμβάνουν υψηλή σημασιολογική πληροφορία σε αντίθεση με τα πιο ρηχά επίπεδα που ενέχουν τοπικό χαρακτήρα. Παρόμοια λογική εκφράζεται και στις έρευνες [28], [19] και [17] που αναφέρονται στις αρχιτεκτονικές των CNN. Ο δεύτερος κλάδος αποσκοπεί μέσω από σειριακά επίπεδα υπολογιστικών πράξεων να εκμεταλλευτεί αυτά τα τοπικά στοιχεία και να τα εγχύσει κατάλληλα στα προηγούμενα υπολογισμένα καθολικά στοιχεία του πρώτου κλάδου.

Η βελτιστοποίηση της πρώιμης πρόβλεψης, που λαμβάνει χώρα στο δεύτερο κλάδο, χρησιμοποιεί ως επί το πλείστον τα δύο πρώτα χαρακτηριστικά του ResNet50 και το πρώτο χαρακτηριστικό του SegFormer. Ως εκ

τούτου, ο εμπλουτισμός της πρώιμης πρόβλεψης επιτυγχάνεται με την εισαγωγή λεπτομερειών σε τοπικό πλέον επίπεδο, ώστε να παραχθεί μια φινιρισμένη και λεπτομερέστερη πρόβλεψη. Με την ανάμιξη των τριών αυτών στοιχείων δημιουργείται το στοιχείο με την ονομασία FB (Fused Bottom), όπως φαίνεται παρακάτω:

$$X_1X_2 = BConv(Concat(X_1,\, X_2)) \tag{10}$$

$$C_1X_1 = BConv(C_1 \odot X_1) \tag{11}$$

$$C_1X_1X_2 = BConv(Inter(BConv(Concat(C_1X_1 \odot X_1X_2,\, X_1X_2)))) \tag{12}$$

$$FB = BConv(Concat(MF \odot C_1X_1X_2,\, MF)), \tag{13}$$

όπου είναι εμφανές πως χρησιμοποιείται εκ νέου το ενδιάμεσο χαρακτηριστικό MF που υπολογίστηκε στην εξίσωση (1).

Εκτός από το στοιχείο MF, υπάρχει ακόμα ένα ισχυρό χαρακτηριστικό με ιδιότητες κυρίως καθολικές. Πρόκειται για το στοιχείο C_4C_3, όπως αυτό υπολογίστηκε από την εξίσωση (4). Το C_4C_3 περιλαμβάνεται σε αυτό το σημείο του δικτύου μας ώστε να εισάγει ακόμη περισσότερη εννοιολογική γνώση. Η τελική πρόβλεψη παράγεται όπως φαίνεται παρακάτω:

$$F1 = C_4C_3C_2MF + FB \tag{14}$$

$$F2 = ChannelAttn(F1) \odot F1 \tag{15}$$

$$F3 = BConv(Concat(C_4C_3,\, FB)) \tag{16}$$

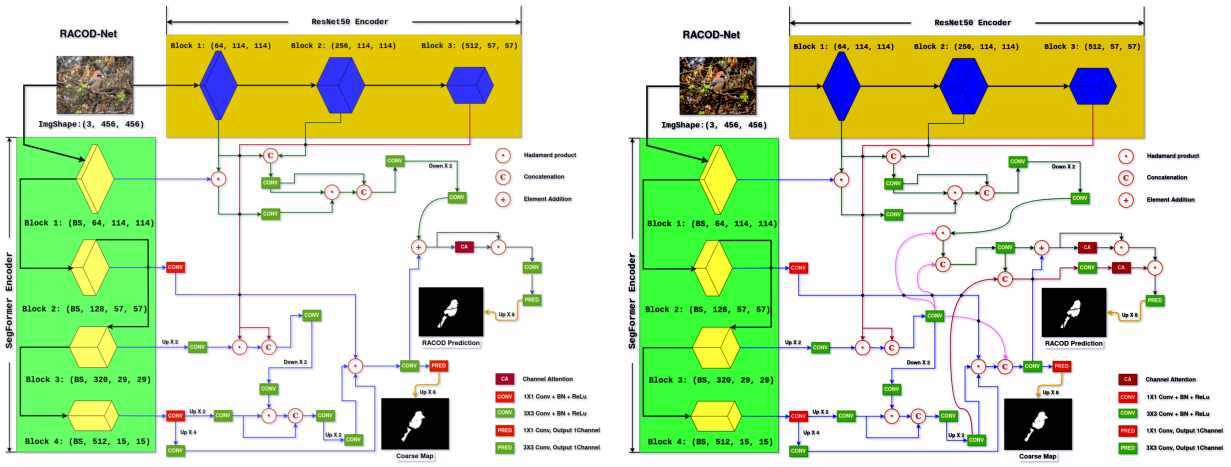
$$F4 = ChannelAttn(F3) \odot F2 \tag{17}$$

$$RefinedMap = Inter(Pred(F4)) \tag{18}$$

Άξιο αναφοράς αποτελεί η χρήση της άθροισης στην εξίσωση (14). Παρόμοια με τη αξιοποιηση των συνδέσεων παράκαμψης που αναπτύσσει η αρχιτεκτονική ResNet [18] εκτελείται και η άθροιση των ενδιάμεσων στοιχείων $C_4C_3C_2$MF και FB. Ο κύριος λόγος έγγυται στην ανάγκη να μεταβιβάσουμε υψηλού επιπέδου σημασιολογικές πληροφορίες αμετάβλητες στα τελευταία επίπεδα του νευρωνικού μας δικτύου. Εκτός από τα παραπάνω, γίνεται χρήση και ενός επιπέδου με την ονομασία Channel Attention, το οποίο εισήχθη από την έρευνα [65], με στόχο να επικεντρωθεί το αποτέλεσμα στα σημαντικότερα χαρακτηριστικά του αντικειμένου. Κάθε κανάλι στον αποκωδικοποιητή μας αποτελεί και κάποιο πιθανό χαρακτηριστικό του αντικειμένου της αρχικής εικόνας. Μέσω του συγκεκριμένου επιπέδου δίνεται σημασία μόνο στα σημαντικά κανάλια του μοντέλου μας. Ο δεύτερος κλάδος ολοκληρώνεται με την εξίσωση (18), όπου η τελική πρόβλεψη δειγματοληπτείται και αυτή οχτώ φορές προς τα άνω ώστε να ταιριάξει σε ανάλυση με την επαληθευμένη αλήθεια κατά τη διάρκεια της εκπαίδευσης.

Περαιτέρω επεξήγηση της αρχιτεκτονικής

Όπως αναφέρθηκε προηγουμένως, η προτεινόμενη αρχιτεκτονική μας συγχωνεύει με ένα μοναδικό τρόπο τα ρηχότερα στρώματα που παράγονται από τους κωδικοποιητές μας. Αυτή η τεχνική τελικά εξάγει ένα ενδιάμεσο προϊόν που ονομάζεται FB (Fused Bottom), όπως υπολογίζεται από την εξίσωση (13). Ένας τέτοιος τανυστής πληροφοριών χαμηλού επιπέδου πρέπει να χειριστεί κατάλληλα πριν συμβεί περαιτέρω συγχώνευση με τους υψηλού σημασιολογικούς τανυστές μας, που υπολογίζονται από τον πρώτο κλάδο του αποκωδικοποιητή μας. Όπως φαίνεται στο Σχ. 4, αρχικά υπήρχε μια αφελής υλοποίηση του RACOD-Net.



Απλοϊκή Υλοποίηση RACOD-Net

Σχήμα 4: Οπτική διαφορά ανάμεσα στην απλοϊκή και στην τελική αρχιτεκτονική μας.

Η αφελής πρωταρχική ιδέα βασίστηκε και αυτή με τη σειρά της στο αρχικό μας κίνητρο. Για την αποτύπωση τόσο του τοπικού όσο και του παγκόσμιου πλαισίου φαινόταν αρκετά απλό να προχωρήσουμε με μια απλή προσθήκη μεταξύ των τανυστών πληροφοριών χαμηλού και υψηλού επιπέδου, όπως φαίνεται στην αριστερή εικόνα του Σχ. 4. Ωστόσο, αυτή η απλή προσθήκη θα είχε ως αποτέλεσμα μια μέτρια τελική πρόβλεψη που παρουσίαζε μικρές διαφορές από την πρώιμη πρόβλεψη. Αυτοί οι δύο τανυστές αντιπροσωπεύουν διαφορετικές φιλοσοφίες και τεχνολογίες, παρόλο που έχουμε διαρρεύσει ορισμένες τοπικές ιδιότητες στο παγκόσμιο πλαίσιο και το αντίθετο, συγχωνεύοντας το X_2 με τον SegFormer και το C_1 με τον ResNet50. Επομένως, η εκτέλεση μιας τέτοιας προσθήκης δεν συνιστάται, καθώς αυτές οι δύο προβλέψεις έχουν πολύ λίγες ομοιότητες και τα αντίστοιχα βάρη τους αποτελούνται από εξαιρετικά αποκλίνουσες τιμές.

Η θεμελιώδες αυτή αρχική ιδέα άλλαξε και βελτιώθηκε, με βάση την πρωτότυπη αρχιτεκτονική ResNet όπου οι συνδέσεις παράκαμψης προσθέτουν τα βάρη από το ένα στρώμα στο άλλο. Αυτά τα διαδοχικά βάρη μοιράζονται κοινές εννοιολογικές πληροφορίες, καθώς είναι το αποτέλεσμα γειτονικών στρωμάτων συνελίξεων. Μια τέτοια προσθήκη δεν θα ήταν αποτελεσματική εάν αυτά τα βάρη είχαν δημιουργηθεί από στρώματα που απέχουν μεταξύ τους πολλά επίπεδα συνελίξεων. Λαμβάνοντας υπόψη μία τέτοια ανάλυση, ήταν υποχρεωτικό να μειωθεί το σημασιολογικό χάσμα μεταξύ των προβλέψεων του αποκωδικοποιητή προτού πραγματοποιηθεί οποιαδήποτε περαιτέρω ενέργεια.

Προς αυτόν τον στόχο, επισημαίνονται δύο συγκεκριμένα υβριδικά χαρακτηριστικά για τη μείωση των διαφορών που αναφέρθηκαν προηγουμένως. Το ενδιάμεσο παράγωγο MF, όπως υπολογίζεται από την εξίσωση (1) είναι το πρώτο υποψήφιο για μία τέτοια διεργασία. Ο κωδικοποιητής SegFormer εκτελεί πολλαπλές επαναλήψεις αυτοπροσοχής για κάθε χαρακτηριστικό που δημιουργείται. Πιο συγκεκριμένα, το C_3 απαιτεί είκοσι εφτά επαναλήψεις αυτοπροσοχής. Αργότερα, το C_3 αναμιγνύεται με το X_2 παράγοντας το ενδιάμεσο προϊόν που ονομάζεται MF. Δεδομένου ότι το MF περιέχει τόσο καθολικές όσο και τοπικές ιδιότητες, όταν πολλαπλασιάζεται και συνδυάζεται με την πιο ρηχή συγχώνευση των χαρακτηριστικών C_1, X_1 και X_2 , συμβάλλει στο πρώτο βήμα για τη μείωση της σημασιολογικής απόστασης μεταξύ των δύο κλάδων. Στη συνέχεια, η αναφερόμενη άθροιση λαμβάνει χώρα και διέρχεται από ένα στρώμα Channel Attention που εξάγει με τη σειρά του το χαρακτηριστικό F2, σύμφωνα με την εξίσωση (15). Ωστόσο, το χαρακτηριστικό F2 δεν είναι αρκετά ικανό να προωθηθεί ως η τελική εκλεπτυσμένη ακριβής πρόβλεψη. Από την οπτική επιθεώρηση των αποτελεσμάτων μας, η εκπαίδευση του μοντέλου μας με το F2 ως τελικό αποτέλεσμα ξεπερνά κατά πολύ την πρωταρχική αφελή υλοποίηση. Η επίτευξη πιο ακριβών και ισχυρών δυαδικών χαρτών δείχνει ότι αυτό είναι το σωστό μονοπάτι όταν συνδυάζονται διαφορετικές τεχνολογίες στους κωδικοποιητές. Ωστόσο, κατά την οπτική επιθεώρηση των αποτελεσμάτων μας ανακαλύψαμε ότι, παρά την επίτευξη λεπτομερών ορίων και περιορισμένου θορύβου, υπήρχαν ορισμένες προβλέψεις που ταξινομούσαν εσφαλμένα τα μη καμουφλαρισμένα αντικείμενα ως καμουφλαρισμένα.

Ακολουθώντας την ίδια τακτική που μας δίδαξε το χαρακτηριστικό MF, σκεφτήκαμε εάν θα μπορούσαμε να

βελτιώσουμε περαιτέρω την πρόβλεψή μας και να μειώσουμε τις ανακρίβειές της. Τέτοια λάθη απαιτούν περισσότερη έγχυση σημασιολογικής γνώσης στο χαρακτηριστικό F2. Το C_4C_3, όπως υπολογίζεται από την εξίσωση (4), είναι ο δεύτερος υποψήφιος για την ολοκλήρωση της τελικής μας αρχιτεκτονικής. Το C_4C_3 είναι το τελευταίο και βαθύτερο χαρακτηριστικό από το SegFormer που αντλεί πληροφορίες από ολόκληρη την εικόνα. Καταγράφει μακρινές σημασιολογικές συνάφειες μεταξύ κρίσιμων τμημάτων της δεδομένης εικόνας, καθώς περιέχει όλες τις προηγούμενες επαναλήψεις αυτοπροσοχής που έχουν εκτελεστεί. Το F2 στερείται της απαραίτητης ερμηνείας των αντικειμένων και το C_4C_3 μπορεί να χειριστεί αυτήν την απαίτηση. Ωστόσο, για άλλη μια φορά πριν πολλαπλασιαστούν αυτά τα στοιχεία, η σημασιολογική τους απόσταση πρέπει να μειωθεί. Όπως φαίνεται από τις εξισώσεις (16), (17) και το Σχ. 2, προχωράμε συνελίσσοντας τη συγχώνευση του C_4C_3 με το FB και μετέπειτα ακολουθεί ένα επίπεδο Channel Attention. Τέλος, το παραγόμενο χαρακτηριστικό F4 μπορεί να πολλαπλασιαστεί με το F2 για να μειώσει τυχόν εναπομένοντα σημασιολογικά κενά, σχηματίζοντας τον τελικό εμπλουτισμένο τανυστή. Αυτός ο τανυστής παρακολουθεί εξαρτήσεις μεγάλης εμβέλειας, συγκεντρώνει το παγκόσμιο πλαίσιο και διατηρεί αποτελεσματικά τις απαραίτητες τοπικές συνελικτικές ιδιότητες.

Υπάρχει μια ακόμη σημαντική διαφορά μεταξύ των υλοποιήσεών μας. Η αφελής ανάπτυξη του RACOD-Net αποφεύγει την παραγωγή του ενδιάμεσου προϊόντος που ονομάζεται $C_4C_3C_2MF$, όπως υπολογίζεται από την εξίσωση (8) και φαίνεται στο Σχ. 4. Εφόσον, το χαρακτηριστικό MF επηρεάζει τον δεύτερο κλάδο του αποκωδικοποιητή μας, όπως περιγράφηκε προηγουμένως, επιλέγουμε στην τελική έκδοση της αρχιτεκτονικής μας να αυξήσουμε τη συνεισφορά του και στον πρώτο κλάδο. Στη συνέχεια, μεταβιβάζεται σε ένα στρώμα συγχώνευσης, σύμφωνα με την εξίσωση (8), και αναπτύσσει ένα νέο συγκεντρωτικό χαρακτηριστικό το οποίο ύστερα συμμετέχει στην αναφερόμενη άθροιση της εξίσωσης (14). Οι δύο κλάδοι του αποκωδικοποιητή μας είναι προσαρμοσμένοι με συγκεκριμένο τρόπο για να ταιριάζουν σημασιολογικά ο ένας με τον άλλον. Όχι μόνο ο δεύτερος κλάδος είναι σημασιολογικά κοντά στον πρώτο κλάδο μέσω της χρήσης του στοιχείου MF, αλλά και μέσω του $C_4C_3C_2MF$ ο πρώτος κλάδος τείνει εννοιολογικά προς τον δεύτερο.

Όλο αυτό το κίνητρο της σύντηξης διαφορετικών κωδικοποιητών, ενός CNN και ενός Transformer, απαιτεί μια αμφίδρομη αλληλεπίδραση μεταξύ τους. Η αδρή προσθήκη ανάμεσά τους οδηγεί σε αποτελέσματα μη αποδεκτά. Η επιλογή της διαδρομής, όπως περιγράφτηκε νωρίτερα, και η προτίμηση αυτών των δύο υβριδικών χαρακτηριστικών, MF και $C_4C_3C_2MF$, για την εκτέλεση όλων των συγκεντρωτικών διεργασιών απαιτεί μία βαθύτερη κατανόηση των διαθέσιμων τεχνολογιών των δύο κωδικοποιητών μας.

Συνάρτηση Απώλειας

Η συνάρτηση απώλειας του RACOD-Net δίνεται από την παρακάτω εξίσωση:

$$L = L_{cm} \quad + \quad L_{rm},$$

όπου η L_{cm} είναι η συνάρτηση απώλειας μεταξύ της πρώιμης πρόβλεψης P1 με την επαληθευμένη αλήθεια και η L_{rm} υπολογίζεται μεταξύ της τελικής πρόβλεψης P2 με την επαληθευμένη αλήθεια.

Η L_{cm} ορίζεται ως εξής:

$$L_{cm} = L^w_{IoU}(P1, GT) \quad + \quad L^w_{BCE}(P1, GT)$$

Η L_{rm} ορίζεται ως εξής:

$$L_{rm} = L^w_{IoU}(P2, GT) \quad + \quad L^w_{BCE}(P2, GT),$$

Κάθε μία από τις δύο αυτές συναρτήσεις, L_{cm} και L_{rm}, προκύπτουν από το άθροισμα δύο επιμέρους συναρτήσεων απώλειας. Η πρώτη από τις δύο είναι η $L^w_{IoU}(\cdot)$ που αποτελεί τον σταθμισμένο λόγο της τομής ως προς την ένωση, δηλαδή ποσοτικοποιεί τον βαθμό επικάλυψης μεταξύ της πρόβλεψης και της επαληθευμένης αλήθειας. Η $L^w_{BCE}(\cdot)$ ορίζεται ως η σταθμισμένη δυαδική απώλεια εντροπίας. Ο σταθμισμένος παράγοντας που αναφέρεται ορίζει σε κάθε pixel ένα βάρος α. Αυτό το βάρος στη δική μας περίπτωση υπολογίζεται με τέτοιο τρόπο, ώστε να επιβραβεύσουμε τη συνάρτηση απώλειας όταν παράγεται μία πρόβλεψη κοντά στην αλήθεια και ταυτόχρονα να επιπλήξουμε τη συνάρτηση απώλειας σε περίπτωση που η παραγόμενη πρόβλεψη απέχει πολύ από την πραγματικότητα.

Σύνολο Δεδομένων

Τα δεδομένα που χρησιμοποιούνται κατά την εκπαίδευση του μοντέλου μας αντλούνται από δύο διαφορετικές πηγές. Από το σύνολο δεδομένων με την ονομασία CAMO [27] χρησιμοποιούμε 1.000 εικόνες, κάθε μία από τις οποίες εγγυάται την ύπαρξη τουλάχιστον ενός καμουφλαρισμένου αντικειμένου. Από το σύνολο δεδομένων με την ονομασία COD10K [11], το οποίο περιλαμβάνει καμουφλαρισμένα αντικείμενα από 78 διαφορετικά σενάρια και περιβάλλοντα, γίνεται χρήση 3.040 εικόνων. Δηλαδή, το μοντέλο μας εκπαιδεύεται σε 4.040 εικόνες. Για την αξιολόγηση του μοντέλου μας, χρησιμοποιούμε 250 εικόνες από το CAMO, 2.026 εικόνες από το COD10K, 76 εικόνες από το CHAMELEON [56] και 4.121 εικόνες από το NC4K [34]. Η χρήση αυτών των δεδομένων συνάδει με τις υπόλοιπες έρευνες όπως οι [11], [70], [31] ώστε τα αποτελέσματα να είναι αντίστοιχα και να μπορούν να συγκριθούν.

Μέθοδοι Αξιολόγησης

Αξιολογούμε το μοντέλο μας με παρόμοιο τρόπο που ακολούθησαν προηγούμενες μελέτες, όπως οι [70], [25], [11], [77], [72]. Χρησιμοποιούμε στο σύνολο τέσσερις μεθόδους αξιολόγησης για κάθε σύνολο δεδομένων, οι οποίες αναφέρονται παρακάτω:

- S-measure (S_m) [7], που αποτελεί μία μέθοδο δομικής αξιολόγησης.
- Weighted F-measure (F_m^w) [35], που περιλαμβάνει έναν συνδυασμό ακρίβειας (precision) και ανάκλησης (recall).
- Adaptive E-measure (αE) [8], που χρησιμοποιείται για την αξιολόγηση της τμηματοποίησης σε επίπεδο τόσο εικονοστοιχείων όσο και σε ολόκληρη την εικόνα.
- Mean Absolute Error (MAE) [42], που αντιπροσωπεύει την απόλυτη διαφορά της πρόβλεψης με την επαληθευμένη αλήθεια.

Λεπτομέρειες Υλοποίησης

Το RACOD-Net υλοποιείται με χρήση της βιβλιοθήκης PyTorch [41]. Τα βάρη των δύο κωδικοποιητών αρχικοποιούνται με προ-εκπαιδευμένα βάρη από το σύνολο δεδομένων ImageNet. Ως βελτιστοποιητής (optimizer) χρησιμοποιείται ο Adam [26] με αρχικό χρόνο μάθησης το 2e-5. Για να τιμωρήσουμε τα μεγάλα βάρη και να τα διατηρήσουμε σε χαμηλές τιμές ορίζουμε την υπερ-παράμετρο weight decay του βελτιστοποιητή ίση με τον αρχικό χρόνο μάθησης. Για τον κωδικοποιητή SegFormer υιοθετούμε την έκδοση Mit-b4, η οποία ρυθμίζει τις απαιτούμενες υπερ-παραμέτρους. Εκπαιδεύουμε το μοντέλο μας για 39 εποχές, εισάγωντας 6 εικόνες κάθε φορά με χωρικές διαστάσεις 456x456 και η διάσταση των καναλιών είναι ίση με 768. Η εκπαίδευση διαρκεί είτε 11 ώρες αν γίνει χρήση της κάρτας γραφικών P100 16Gb είτε 6 ώρες και 30 λεπτά αν γίνει χρήση της κάρτας γραφικών GeForce RTX 3060 12Gb.

Αποτελέσματα

Συγκρίνουμε το μοντέλο μας με δεκαέξι άλλα μοντέλα, τα οποία παρουσιάζουν εξαιρετικά αποτελέσματα. Στον Πίνακα 1 γίνεται αντιληπτό πως το μοντέλο μας ξεπερνάει σε επιδόσεις αρκετές προηγούμενες μελέτες και θέτει τον πήχη στις προβλέψεις σε ποσοστό 62.5%. Έχει τη δυνατότητα να ανιχνεύσει μικρά και μεγάλα καμουφλαρισμένα αντικείμενα, εντοπίζοντας με μεγάλη επιτυχία τα όρια τους. Ο θόρυβος στις περισσότερες περιπτώσεις έχει εξουδετερωθεί και κατά συνέπεια το αποτέλεσμα έχει υψηλό βαθμό ευκρίνειας. Ακόμα και όταν στην εικόνα εισόδου εμπεριέχονται παραπάνω από ένα αντικείμενα ή αντικείμενα τα οποία διαχωρίζονται σε επιμέρους τμήματα, το μοντέλο μας παρουσιάζει άριστα αποτελέσματα, σχεδόν ίδια με την επιβεβαιωμένη αλήθεια. Το γεγονός ότι η απόδοση του μοντέλου μας ξεπερνάει όλες τις προηγούμενες μελέτες σε τόσο υψηλό ποσοστό, αποτελεί ενδεικτικό στοιχείο της επιτυχημένης μας αρχιτεκτονικής.

Method	NC4K				COD10K-Test				CAMO-Test				CHAMELEON			
	4,121 Images				2,026 Images				250 Images				76 Images			
	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$
PraNet$_{2020}$ [12]	.822	.871	.724	.059	.789	.839	.629	.045	.769	.833	.663	.094	.860	.898	.763	.044
SINet$_{2020}$ [11]	.808	.883	.723	.058	.776	.867	.631	.043	.745	.825	.644	.092	.872	.938	.806	.034
SLSR$_{2021}$ [34]	.840	.902	.766	.048	.804	.882	.673	.037	.787	.855	.696	.080	.890	.936	.822	.030
MGL-R$_{2021}$ [73]	.833	.893	.739	.053	.814	.865	.666	.035	.782	.847	.695	.085	.893	.923	.812	.031
PFNet$_{2021}$ [36]	.829	.892	.745	.053	.800	.868	.660	.040	.782	.852	.695	.085	.882	.942	.810	.033
C^2FNet$_{2021}$ [57]	.838	.898	.762	.049	.813	.886	.686	.036	.796	.864	.719	.080	.888	.932	.828	.032
UGTR$_{2021}$ [77]	.839	.886	.746	.052	.817	.850	.666	.036	.784	.859	.794	.086	.888	.921	.794	.031
SINetV2$_{2022}$ [10]	.847	.898	.770	.048	.815	.863	.680	.037	.820	.875	.743	.070	.888	.930	.816	.030
DGNet$_{2022}$ [24]	.857	.907	.784	.042	.822	.877	.693	.033	.839	.901	.769	.057	.890	.934	.816	.029
SegMaR$_{2022}$ [25]	.841	.905	.781	.046	.833	.895	.724	.033	.815	.872	.742	.071	.897	.950	.835	.027
ZoomNet$_{2022}$ [72]	.853	.907	.784	.043	.838	.893	.729	.029	.820	.883	.752	.066	**.902**	.952	**.845**	**.023**
FDNet$_{2022}$ [77]	.834	.895	.750	.052	.837	.897	.731	.030	.844	.903	.778	.062	.894	.948	.819	.030
TPRNet$_{2022}$ [74]	.854	.903	.790	.047	.829	.892	.725	.034	.814	.870	.781	.076	.891	.930	.816	.031
DTINet$_{2022}$ [31]	.863	**.915**	.792	**.041**	.824	.893	.695	.034	.857	.912	.796	.050	.883	.928	.813	.033
CamoFormer-S$_{2022}$ [70]	**.888**	**.941**	**.840**	**.031**	**.862**	**.932**	**.772**	**.024**	**.876**	**.935**	**.832**	**.043**	.891	**.953**	.829	.026
CamoFormer-P$_{2022}$ [70]	**.892**	**.941**	**.847**	**.030**	**.869**	**.931**	**.786**	**.023**	**.872**	**.931**	**.831**	**.046**	**.910**	**.970**	**.865**	**.022**
RACOD-Net (Ours)	**.889**	**.939**	**.855**	**.031**	**.872**	**.942**	**.804**	**.022**	**.868**	**.928**	**.835**	**.047**	**.917**	**.971**	**.887**	**.021**

Πίνακας 1: Ποσοτικά αποτελέσματα σε δημόσια σύνολα δεδομένων. Τα αποτελέσματα προηγούμενων μελετών έχουν επιβεβαιωθεί από τα [70], [31], [25], [36] and [24]. Με χρώμα Κόκκινο, Πράσινο, και Μπλε υποδεικνύεται η πρώτη, δεύτερη και τρίτη καλύτερη απόδοση. Τα σύμβολα '↑/↓' υποδηλώνουν αντίστοιχα πως όσο υψηλότερο/χαμηλότερο είναι το ποσοτικό αποτέλεσμα τόσο καλύτερο είναι.

Πειράματα

Εκτελούμε διάφορα πειράματα με απώτερο σκοπό να αιτιολογήσουμε τις επιλογές που υλοποιήσαμε στην αρχιτεκτονική μας. Αφαιρώντας διάφορα στοιχεία ή τροποποιώντας κάποιες υπερ-παραμέτρους, λαμβάνουμε συγκεκριμένα ποσοτικά αποτελέσματα, τα οποία οδηγούν σε προβλέψεις χαμηλότερου επιπέδου, επιβεβαιώνοντας τις τελικές μας προτιμήσεις. Συγκεκριμένα, εκτελούμε τα παρακάτω πειράματα:

- Από την τελική μας αρχιτεκτονική πραγματοποιείται αφαίρεση της τελικής πρόβλεψης και διατήρηση μόνο της αρχικής πρώιμης. Δηλαδή, ο αποκωδικοποιητής μας πλέον περιέχει μόνο έναν κλάδο, τον πρώτο κλάδο.
- Χρήση της απλοϊκής υλοποίησης του RACOD-Net, όπου πραγματοποιείται μονάχα μία αδρή άθροιση με σκοπό την ένωση των δύο κλάδων του αποκωδικοποιητή. Η απλοϊκή αυτή αρχιτεκτονική απεικονίζεται στην αριστερή εικόνα του Σχ. 4.
- Μείωση της διάστασης των καναλιών σε 256, 128 και 64.
- Μείωση των χωρικών διαστάσεων της εικόνας εισόδου από 456x456 σε 256x256.

Τα αποτελέσματα των παραπάνω πειραμάτων απεικονίζονται στον Πίνακα 2.

Settings	NC4K				COD10K-Test				CAMO-Test				CHAMELEON			
	4,121 Images				2,026 Images				250 Images				76 Images			
	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$
Πρώιμη Πρόβλεψη	.886	.931	.834	.035	.870	.918	.778	.025	.865	.917	.812	.052	.913	.949	.857	.026
Απλοϊκή Υλοποίηση	.872	.928	.837	.037	.860	.938	.792	.024	.847	.915	.815	.055	.902	.957	.872	.025
Διάσταση καναλιών: 256	.884	.936	.845	.033	.868	.942	.796	.022	.861	.919	.823	.050	.910	.960	.870	.024
Διάσταση καναλιών: 128	.886	.936	.842	.033	**.872**	.937	.794	.023	.864	.925	.825	.049	.915	.960	.874	.022
Διάσταση καναλιών: 64	.887	.934	.837	.034	.868	.922	.779	.024	.866	.922	.820	.050	.908	.944	.855	.027
Εικόνα Εισόδου: 256x256	.848	.916	.800	.044	.834	.928	.751	.029	.801	.870	.746	.071	.850	.921	.791	.038
RACOD-Net	**.889**	**.939**	**.855**	**.031**	**.872**	**.942**	**.804**	**.022**	**.868**	**.928**	**.835**	**.047**	**.917**	**.971**	**.887**	**.021**

Πίνακας 2: Ποσοτικά αποτελέσματα από τα πειράματά μας. Τα καλύτερα αποτελέσματα είναι σημειωμένα με έντονη γραφή. Τα σύμβολα '↑/↓' υποδηλώνουν αντίστοιχα πως όσο υψηλότερο/χαμηλότερο είναι το ποσοτικό αποτέλεσμα τόσο καλύτερο είναι.

Σε αυτό το σημείο οφείλουμε να επισημάνουμε μία σημαντική παρατήρηση που προκύπτει από τον Πίνακα 2. Η απλοϊκή υλοποίηση της αρχιτεκτονικής μας επιτυγχάνει σχεδόν τα ίδια αποτελέσματα με το πρώτο μας πείραμα, όπου χρησιμοποιήσαμε μόνο την πρώιμη πρόβλεψη της τελικής μας αρχιτεκτονικής. Υπό αυτές τις συνθήκες αξιολόγησης, γίνεται εύκολα αντιληπτό πως η θεωρητικά τελική φινιρισμένη πρόβλεψη από την απλοϊκή υλοποίηση είναι εξίσου μέτρια με την πρώιμη πρόβλεψη της προτεινόμενης αρχιτεκτονικής. Μέσω αυτής της διαδικασίας έχουμε επαληθεύσει και με ποσοτικά δεδομένα ότι κατά τη σύντηξη διαφορετικών χαρακτηριστικών από ένα συνελικτικό νευρωνικό αποκωδικοποιητή και έναν Transformer κωδικοποιητή δεν αρκεί μια απλή προσθήκη των παραγόμενων τανυστών. Οφείλουμε να μειώσουμε το σημασιολογικό χάσμα των χαρακτηριστικών που εξάγονται από δύο τόσο διαφορετικούς κωδικοποιητές προτού προχωρήσουμε σε μία περαιτέρω ανάμιξη τους.

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Σε αυτή την ερευνητική και πειραματική διαδικασία εισηγάγαμε μία νέα και πολλά υποσχόμενη αρχιτεκτονική, η οποία παρουσιάζει εντυπωσιακά αποτελέσματα. Όλοι οι συνδυασμοί και οι συγχωνεύσεις των χαρακτηριστικών της εισαγόμενης εικόνας, καθώς και οι υπερ-παράμετροι του μοντέλου μας, μελετήθηκαν προσεχτικά και μεθοδικά ώστε να δομηθεί η συγκεκριμένη αρχιτεκτονική.

Υπάρχουν αρκετές περιοχές της όρασης υπολογιστών στον τομέα της ιατρικής επιστήμης, όπως είναι η τμηματοποίηση πολύποδων και όγκων. Και στις δύο αυτές περιπτώσεις το χρώμα, το σχήμα και η υφή των υπό εξέταση παθογενειών ταιριάζουν αρκετά με τους τριγύρω υγιείς ιστούς και θυμίζουν αρκετά τις ιδιότητες των καμουφλαρισμένων αντικειμένων. Αξιολογήσαμε το μοντέλο μας και σε σύνολα δεδομένων που αφορούν την τμηματοποίηση πολύποδων. Τα αποτελέσματα της αξιολόγησης είναι αρκετά ενθαρρυντικά, καθώς σε κάποια σύνολα δεδομένων το μοντέλο μας παράγει εξαιρετικές προβλέψεις, και υποδεικνύουν πως η παρούσα αρχιτεκτονική έχει τη δυνατότητα περαιτέρω γενίκευσης και εφαρμογής.

Εκτός από τον τομέα των καμουφλαρισμένων αντικειμένων, αποσκοπούμε στο να εκπαιδεύσουμε το μοντέλο μας και σε άλλα διαφορετικά σενάρια της όρασης υπολογιστών. Υπάρχουν αρκετές μελέτες και σύνολα δεδομένων για αρκετούς τομείς και ως εκ τούτου θα ήταν αρκετά ενδιαφέρον να παρατηρήσουμε την επίδοση και τη συνεισφορά του μοντέλου μας σε αυτές τις συνθήκες. Η ανίχνευση προεξέχοντος αντικειμένου (salient object detection), η ανίχνευση μικρών αντικειμένων (small object detection), η ανίχνευση αντικειμένων μέσω βίντεο (video object detection) και η ανίχνευση αντικειμένων από εναέρια μέσα (object detection in aerial images) θα μπορούσαν να είναι πιθανές εφαρμογές για την επέκταση της αρχικής αρχιτεκτονικής του RACOD-Net. Ωστόσο, όταν πρόκειται να αυξηθεί το μέγεθος του συνόλου δεδομένων μας, οφείλουμε να τροποποιήσουμε κατάλληλα την αρχιτεκτονική μας ώστε να είναι σε θέση να ανταπεξέλθει, χωρίς να αυξηθούν οι απαιτούμενοι υπολογιστικοί πόροι. Δηλαδή, το μέγεθος της παρτίδας που εισάγεται είναι επιβεβλημένο να αυξηθεί αρκετά με κόστος την τροποποίηση κάποιας άλλης παραμέτρου ή κάποιου τμήματος του δικτύου μας.

Ελπίζουμε πως το RACOD-Net θα μπορέσει να λειτουργήσει ως πλαίσιο αναφοράς που θα διεγείρει περισσότερες νέες ιδέες σε δύσκολες περιοχές της όρασης υπολογιστών. Ο κώδικάς μας και τα εκπαιδευμένα βάρη, τα οποία

οδηγούν στα αποτελέσματα του Πίνακα 1, βρίσκονται διαθέσιμα στο: `https://github.com/mikestratakis/RACOD-Net`.

Chapter 1

Introduction

1.1 State Of Computer Vision

Computer vision is a field of artificial intelligence that processes visual data by excessively using Machine Learning through Deep Learning techniques. Specifically, over the last decade, deep convolutional neural networks have produced several achievements over the different aspects of computer vision. Through these innovations, it is possible for an artificial model to recognize objects, patterns as the human eye and create semantic understanding of the world as the human mind.

State-of-the-art computer vision applications, despite having heavy parameter architectures, have been successfully trained by leveraging graphical processing units (GPUs) for parallel computing implementations, resulting in faster processing of data and systems able to achieve high precision and minimal false predictions [39]. Major products that are commonly used every day are utilizing core concepts of computer vision techniques, thus demonstrating an important role in object detection and semantic segmentation in healthcare and manufacturing domains.

Computer Vision as a field of research conceals several challenges, mainly because of the human eye being too flawless in many visual tasks [22]. For an artificial model to achieve similar performance a vast amount of computations and visual data are required. To further enhance the outcome and minimize over fitting such models require architectures than can obtain rich information and high-level understanding of the provided data.

1.2 Thesis Motivation

One of the key abilities of the human eye relies on the successful and instant object recognition and semantic realization of the world. Under several conditions, the identification of objects and their understanding is accurate and the process comes naturally and effortlessly. On the contrary, computer vision models require to extract a set of features from the given digital representation of the world and later analyze them.

However, there are objects either natural or artificial made with the capability of being camouflaged, thus avoiding salient features to be extracted. The evolution of animals in the wild and the human-made camouflaged strategies reduces the probability of detection or recognition by potentials predators or enemies [55]. In addition, the evolution of new diseases derives a complex and challenging area in modern healthcare industry detection of lung infections and their boundaries from medical images [45]. A commonly used strategy by these objects is to adapt the pattern, color and other morphological properties of the surrounding environments to such a degree so that the driven correlation in appearance will lead even the experienced human eye in a false decision [43]. Overall, camouflage manipulates the visual representation that reaches the viewer and highly increases the challenges of an accurate segmentation [37].

Despite these challenges, there have been several studies based on deep learning architectures that have shown a promising performance, even with a small amount of data available. Many researchers have based

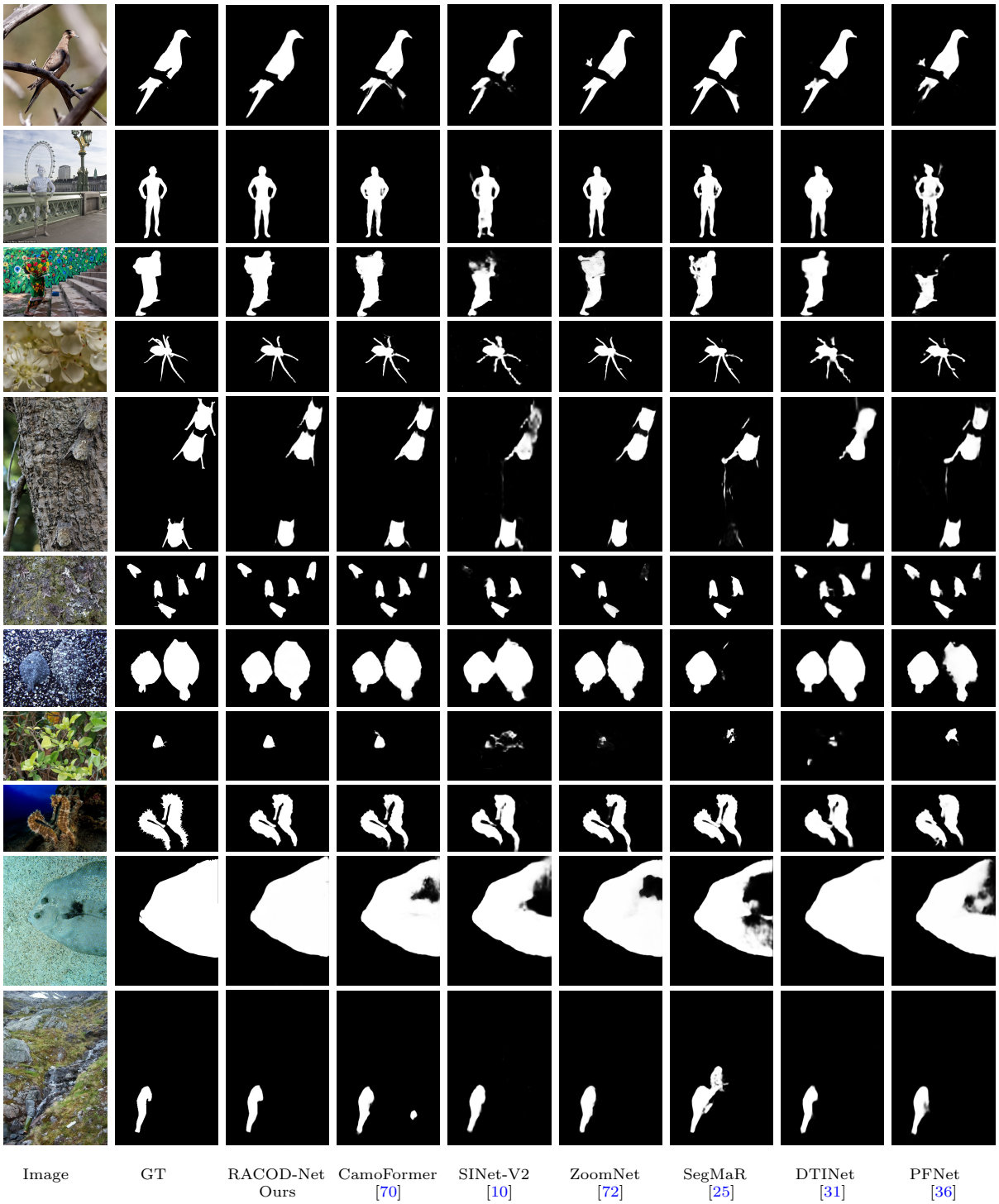


Figure 1.1: Visual comparisons among challenging objects, from different and competitive models.

their models in complex convolutional neural networks and recently in complex Transformer neural networks. However, the produced segmentation results are often mediocre and unsatisfactory, especially in cases where the object is either perfectly blended with the surrounding or neighbours with several other camouflaged objects.

A fine-grained recognition model is expected to outcome such challenges and ultimately produce effective results, aiming to overcome the limitations and faults born from previous works. Achieving such an ob-

jective not only enhances the effectiveness of computer vision, but also generates greater profits in image segmentation and object detection.

1.3 Thesis Contribution

Transformer architectures have revolutionized the way attention has been implemented in modern deep learning models by drawing global long-range dependencies between input sequence elements [60]. On the contrary, convolutional neural networks are able to extract features through convolutional structures that glide through the input sequence based by the dimension of the kernel [40]. These features integrate either low-level information in shallow layers or high-level semantic information in deeper layers [19, 28].

Unlike previous works, in this paper we introduce a novel architecture that focuses on fusing appropriately the low-level information produced by traditional CNNs with the global context derived from a Vision Transformer (ViT) architecture. More specifically, we implement a powerful and effective framework termed Refined Accurate Camouflage Object Detection Network (**RACOD-Net**) that utilizes two backbone encoders and an state-of-the-art decoder that handles the described fusion operation.

Our backbone consists of the Convolutional ResNet50 encoder [18], that adopts shortcut connections, and the Transformer SegFormer encoder designed for semantic segmentation [68]. In addition, a cascaded partial decoder module is manufactured to successfully combine the information constructed from the encoders, resulting in an enriched segmentation outcome.

As shown in Fig.1.1, our contribution, RACOD-Net, outperforms several previous studies over challenging objects and scenarios.

Chapter 2

Related Work

2.1 Object Detection

Object detection is a challenging aspect of Computer Vision and its main objective relies on drawing a bounding box for every object of interest and at the same time assign them a class label [3]. Each model expects as an input an image or even a live footage and outputs one or more bounding boxes of certain classes for each predicted object.

One of the first works that introduced a successful method, by achieving a 30% improvement over previous studies, was the R-CNN architecture [16], proposed back in 2014. The concept idea was to extract features through convolutional layers, using a selective search algorithm that would reduce the vast amount of category-based regions to approximately 2000. These regions would represent possible object instances from the given input, such as car, airplane etc, and each feature would have a fixed vector length of 4096 that later would be fed in linear SVMs. Finally, the region with the higher intersection-over-union (IoU) score would be selected. Despite achieving excellent performance at that time, the main drawback of the model was the lack of learning during the selective search state and the slow object detection time of 47 seconds per image [15], that made the model unable to be embedded in real time applications.

As a result, Fast R-CNN was later indroduced by the same author with the aim of enhancing speed and effectiveness [15]. The key concept was based on feeding the input image in the convolutional layers for a feature map to be produced, instead of feeding the proposed regions. From the produced feature maps, using a selective search method region proposals are identified and passed into a region of interest (RoI) pooling layer, where max pooling reshapes any valid regions of interest into a fixed size (H $\times$ W). A region of interest is specified by its top-left corner, its class, its height and its width. Finally, every RoI vector is forwarded into a softmax layer, that holds the class probability and the offset values for the predicted bounding boxes.

R-CNN and Fast-CNN shared one common portion of their architecture, the selective search algorithm that would cost around 2 seconds when the model was implemented in CPU's [49]. This led to Faster R-CNN that included a major algorithmic change where region proposals were calculated through another convolutional network, named Region Proposal Networks (RPNs), reducing the current stage cost from 2 seconds to approximately 10ms per image. The RPNs implemented the idea of attention in neural networks and guided the Fast R-CNN component to detect the objects. More specifically, the last convolutional layer produces a feature map that is later fed into a rectangular sliding window of size n $\times$ n, where n = 3. Each sliding window contains k possible region proposals where each proposal is dependent over k anchor boxes. These anchor boxes centered in the middle of each sliding window pinpoint objects of various sizes and aspect ratios. While the network slides through the feature map pixels, it validates whether these k anchors include objects from the actual image and updates the anchor coordinates.

Unlike R-CNN, Fast R-CNN and Faster R-CNN that utilized regions to detect objects over the input image, YOLO algorithm (You Only Look Once) is based on regression, outputs bounding boxes and class labels concurrently and has the potential of being used in real time applications since the base architecture runs at 45 frames per second [46]. YOLO splits the given image into an S $\times$ S grid and for each grid cell B bounding

boxes and confidence scores are predicted, indicating how accurate the model seems to be. For each of the B bounding boxes the network outputs 5 predictions that consist of the (x, y) coordinates of the object, its height, its width and its class probability. Although YOLO outperformed former studies, due to the spatial constraints of the algorithm it struggled to detect small objects.

Since YOLO had some drawbacks, YOLOv2 [47] and YOLOv3 [48] were later introduced. YOLOv2 made use of anchor boxes and YOLOv3 showed great improvements in detection of smaller objects. Ever since, the YOLO family has shown great ideas, thus enhancing the overall accuracy and smoothness in real-time usage when low-cost components are being used. It is worth mentioning the evolution in computer vision that led from the traditional object-detection algorithms, relied on extracting handcrafted features like Histogram of Oriented Gradients (HOG) [4], to the Two-Stage Object Detectors, like the R-CNN family, who firstly produce regions of proposals and afterwards make predictions for each region and later to the Single-Stage Object Detectors who apply the detection head directly on the feature map, like the YOLO family.

Although these methods showcase great improvements and results, they aim to detect salient objects. Objects with an increased contrast compared to their surroundings that attract human attention. On the contrary, camouflaged objects require an entire new pipeline rendering SOD (Salient Object Detection) techniques less sensitive to provide adequate outcomes [79].

2.2 Image Segmentation

As the term suggests image segmentation is the process of partitioning an image into multiple segments. There are two major types of image segmentation, semantic segmentation were all objects belonging to the same type are assigned the same class label and instance segmentation were all instances of a type are assigned their own separate label.

Recent deep learning methods designed for semantic segmentation make use of the classic encoder-decoder and fully convolutional architecture, as showcased in FCNs [52] where the main idea uses VGG16 as the backbone encoder to produce low resolution features and the decoder is responsible to upsample the output to match the original image. However, the decoder fuses shallow and deeper features by element wise addition to exploit local predictions that respect the global structure. In a similar manner, SegNet [1] uses a decoder network where the feature maps are upsampled using the memorized max pooling indices from the corresponding encoder layer, DeconvNet [38] uses a similar upsampling techique termed unpooling and includes fully-connected layers increasing the computational cost. U-Net [50], designed for biomedical images, transfers the entire feature maps from the encoder to the decoder and concatenates them before further convolutions are performed. UNet++ [78] comes with a new more powerful architecture, based on the original U-Net, by reducing the semantic gap between the encoder and the decoder features before the fusion leading to an easier optimization problem.

Ever since the paper "Attention is all you need" [60] Transformers have been introduced with the concept idea of the attention mechanism to collect global dependencies from the given input. With the great success of Tranformers in NLP tasks ViT architectures were proposed [6], [76]. As shown in figure 2.1, the generic architecture behind ViT relies on splitting the 2-D input image into a sequence of 2-D flattened patches, called tokens, which are mapped to a fixed length linear embedded space. These embedded patches along with a positional embedding are the input to the Transformer encoder. The Multi-head Attention Network inside the encoder helps the model focus on the most important regions. However, one strong drawback lies in the fixed size of the tokens that cannot capture details at different dimensions.

SegFormer [68] is a Transformer model inspired by the ViT architecture that comes with positional-free encodings and a faster self-attention mechanism that contains a sequence reduction process, able to adapt at different resolutions increasing performance and inference time. Swin-Transformer [30], like SegFormer, also introduced hierarchical feature maps and a Shifted Window Self-Attention module to also improve the quadratic complexity of the original ViT architecture.

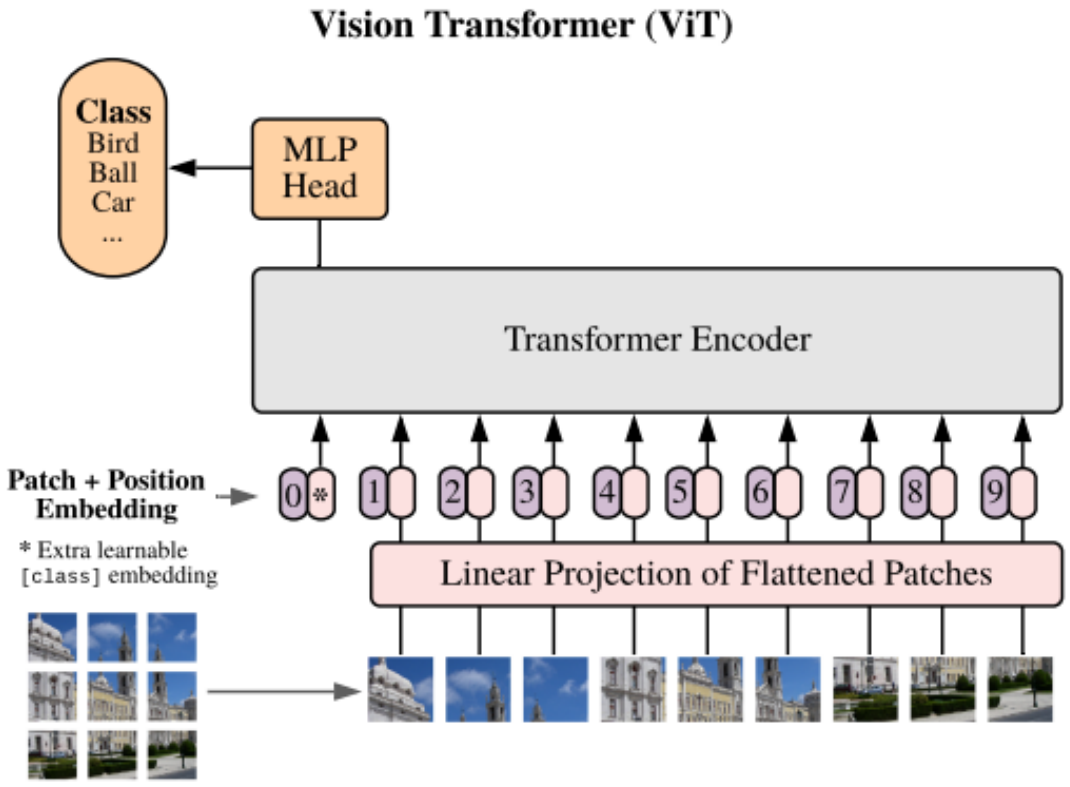


Figure 2.1: The generic ViT architecture [6]

2.3 Camouflage Object Detection

Camouflage object detection relies on segmenting the image in a class-independent manner. Thus, instance-image segmentation is not demonstrated since the task of detection is utilized by assigning the value of 0 to pixels not containing a camouflage object and the value of 1 when a camouflage object is present at each pixel.

Since deep learning evolved, CNNs has showed great results and performance in accomplishing the required task. Great architectures based on CNN backbone like BASNet [44], SINet [11], ANet [27], SINet-V2 [9], BSANet [79], BGNet [58] and PFNet [36] make use of a convolutional backbone encoder and later use a decoder that firstly produces a coarse map output and afterwards through refinement modules the camouflaged object and its boundaries are more accurately and precisely predicted. The concept idea of this partial cascaded decoder having a two stage refinement module is based on the original process represented by [67].

With the evolution of Transformers camouflaged vision tasks have been enhanced as well. A dual-task interactive transformer (DTINet) from [31] utilized two Segformers as their backbone encoder. More specifically, the camouflaged object ground truth (GT) was used to extract features from the foreground stream and the 1-GT to extract features from the background stream. UGTR [69] introduced Bayesian learning into Transformer-based reasoning to successfully capture the uncertainty for camouflaged object detection. CamoFormer [70] adopted a pyramid vision transformer encoder (PVTv2 [62]) and generated a progressive decoder that involved a masked separable attention module to refine the output mask.

Chapter 3

Our Proposal: RACOD Network

3.1 RACOD-Net Architecture

Unlike previous studies RACOD-Net architecture manages to successfully combine two powerful backbone encoders, a CNN encoder and a Transformer encoder, through a novel partial cascaded decoder to output an enriched tensor containing both global and local information. As shown in Fig. 3.1 we initially have a total of 7 features from our backbone encoders. A set of $\{X_k\}_{k=1}^{3}$ features extracted from ResNet50 encoder [18] and a set of $\{C_k\}_{k=1}^{4}$ features extracted from SegFormer encoder [68].

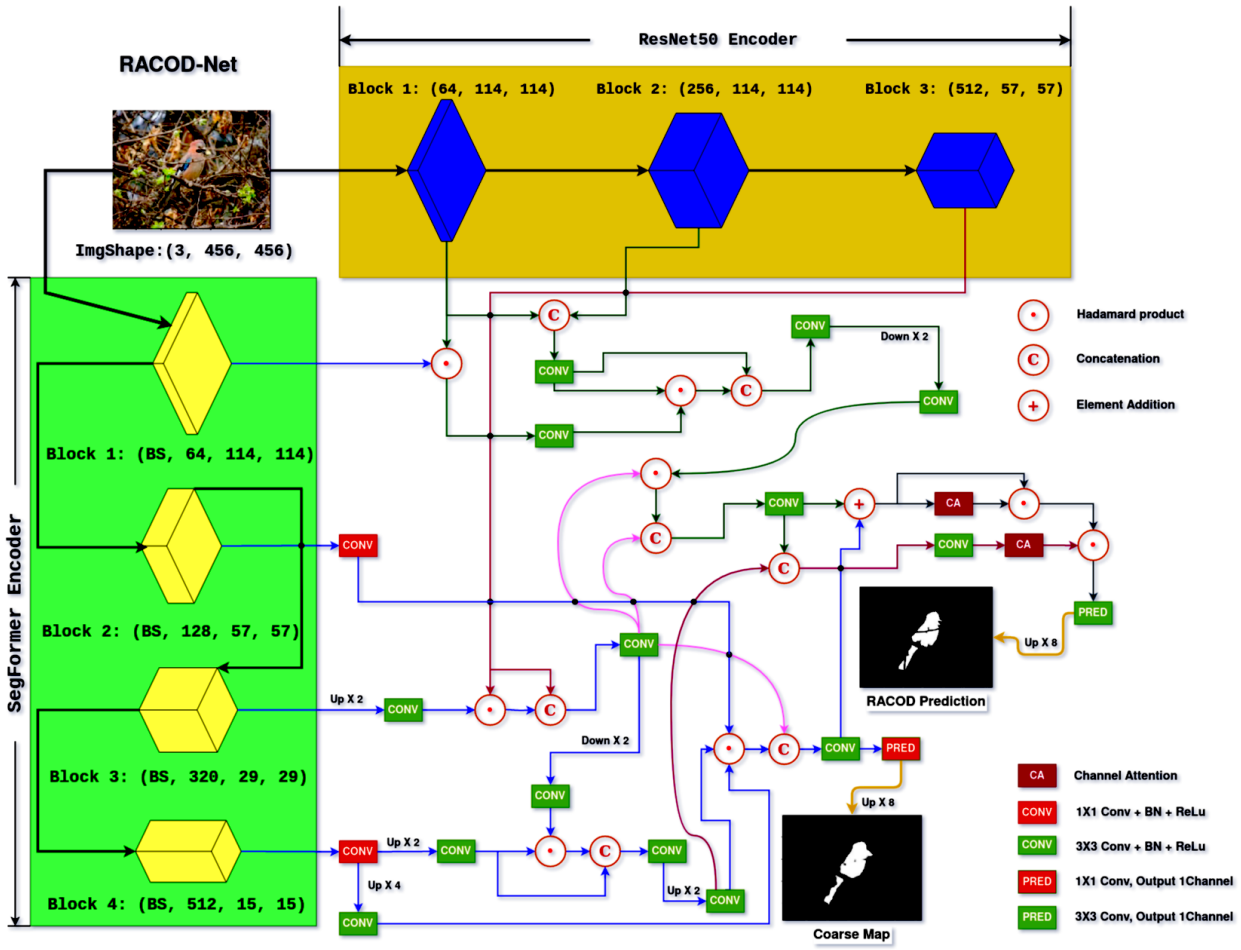


Figure 3.1: The RACOD architecture

3.1.1 CNN Backbone Encoder

The main philosophy behind ResNet50 is based in the VGG-Nets [54]. ResNet50 solves the famous vanishing gradient problem, where the weights remain unchanged as the derivatives vanishes especially in deep networks, through the introduction of shortcuts so the model can propagate larger gradients to initial layers. However, since the main idea is to eventually retain only local information derived from the restricted receptive field of the shallower layers of the ResNet50 architecture we capture only the first 3 layers. Since the ResNet architecture is widely used over numerous papers we choose to avoid further explanation and analysis.

3.1.2 Transformer Backbone Encoder

Additionally, SegFormer acts as a Transformer encoder by initializing an overlapping embedding layer. Thus, given an input image of shape B $\times$ H $\times$ W $\times$ 3 patch merging is performed and afterwards a trainable linear projection layer is applied. The input image is transformed into a sequence of flattened patches with shape of B $\times$ (H $\times$ W) $\times$ C, where C represents the number of channels and B the batch size. The multi-level features produced have shape B $\times$ $\frac{H}{2^{i+1}}$ $\times$ $\frac{W}{2^{i+1}}$ $\times$ C_i, where $i \in 1, 2, 3, 4$, achieving the reduction of spatial dimensions throughout the model. In order to accomplish an overlapping process the first feature is patched with a kernel size of (7, 7), stride of 4 and padding of 3 while the rest features expect a kernel size of (3, 3), stride of 2 and padding of 1. With these specific values the overlapping process is established and therefore among sequential patches the local continuity of information is preserved. Using stride greater than one and less than the kernel size results in decreasing the input's spatial dimension.

The main bottleneck of the original NLP (Natural Language Processing) transformer encoder tasks was the quadratic complexity $\mathcal{O}(N^2)$. This original process could be used with images with low resolution where the images would be split into an input sequence of pixels. However, images with high resolution require even more memory and computations because every pixel of the image has to attend with every other pixel of the image, thus generating a quadratic complexity. To tackle effectively such issues SegFormer introduces a reduction layer just before the self-attention matrix multiplications take place, since the input is forwarded through a convolutional layer with a kernel size of (R, R) and stride of R reducing the complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}\Big(\frac{N^2}{R}\Big)$, where $N = H \times W$. Following this reduction layer the self attention module takes place and is calculated using the traditional transformer formula:

$$Attention(Q,\ K,\ V) = softmax(\frac{QK^T}{\sqrt{d_{head}}})V\,,$$

where Q, K, V are the queries, keys and values of the transformer encoder and d_{head} acts as the value that scales the attention score QK^T.

Unlike the original paper "Attention is all you need" [60] where d_{head} has the same value with the embedding dimension, Segformer sets $d_{head} = \frac{embedding\_dimension}{head}$ and $head = [1, 2, 5, 8]$ for each hierarchical feature respectively.

It is worth mentioning that in our experiments we discovered that for a random tensor of size $(1, 3, 456, 456)$ the execution time of our model fell in half with the use of this reduction layer. RACOD-Net architecture sets reduction ratio R to $[8, 4, 2, 1]$ for each hierarchical feature respectively. This ratio drops half in value for every feature produced due to the fact that deeper features have smaller spatial dimensions compared to shallower features that contain higher spatial dimensions.

Unlike ViT architecture [6] that includes positional embeddings in order for the transformer to remember the order or sequence of the patches, SegFormer skips this step. Intuitively we could argue that for semantic segmentation positional embeddings are not required because from the self attention module we expect only a filtered vector that contains parts of the image that are similar with each other, since we are performing a binary segmentation task instead of a NLP task.

The output from the self attention module x_in is then forwarded into a Mix-FFN (Mix Feed Forward Network), as shown by the formula:

$$x\_out = Mix\_FFN(x\_in) \quad + \quad x\_in\,,$$

where $Mix\_FFN(x\_in) = Linear(GELU(DepthWiseConv_{3x3}(Linear(x\_in))))$.

Since positional embeddings have been skipped, SegFormer uses a depth wise convolution layer inside the Mix-FFN module with a kernel size of (3, 3) that is enough to leak local information while the kernel is sliding through the vector. The SegFormer encoder architecture, as already described, is displayed in Fig. 3.2.

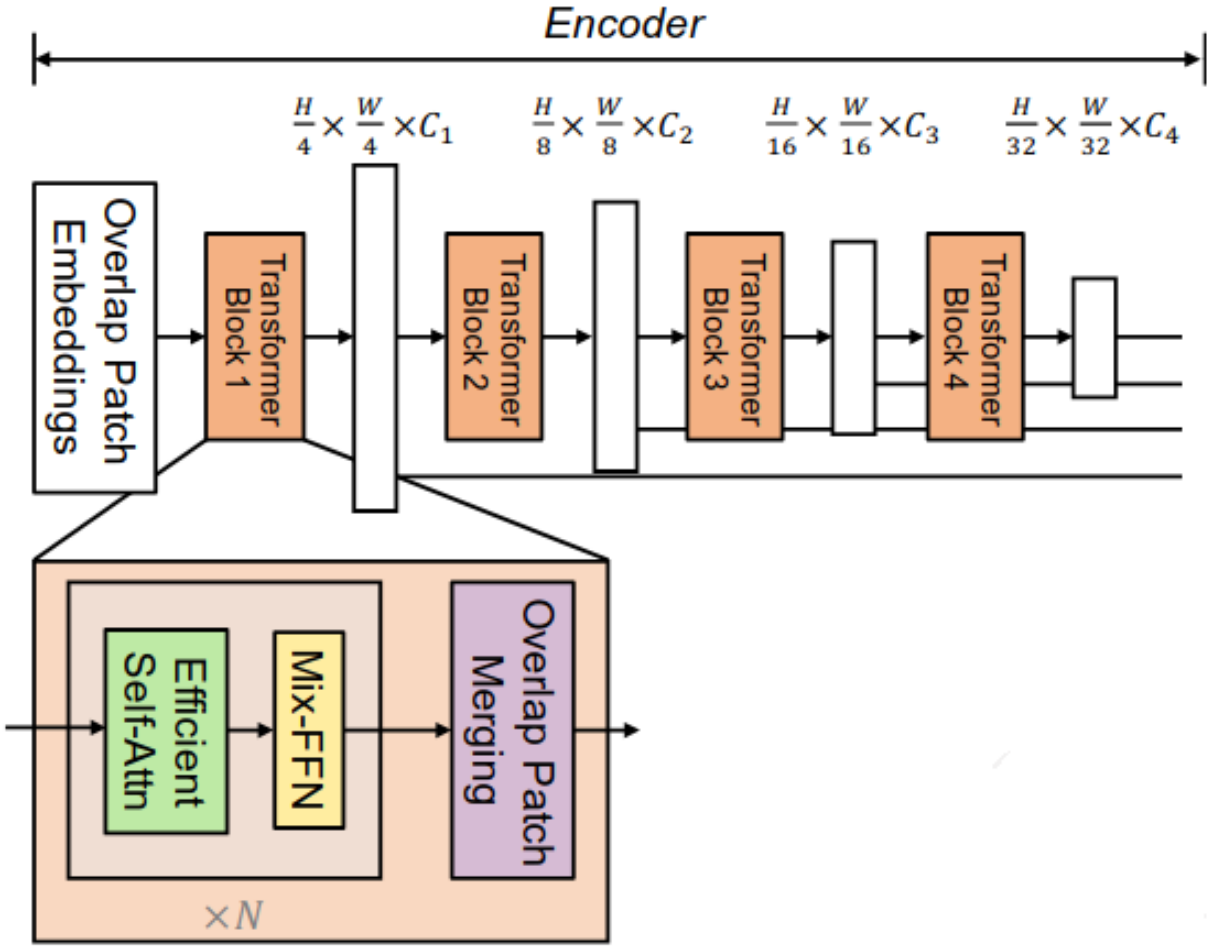


Figure 3.2: The SegFormer Encoder [68]

3.1.3 PRDM: Partial Refined Decoder Module

Although, SegFormer architecture comes with a built-in decoder module RACOD-Net introduces a novel partial cascaded decoder that captures and combines both the features from ResNet50 encoder and SegFormer encoder. The main concept of our cascaded partial decoder is based from the original paper "Cascaded Partial Decoder for Fast and Accurate Salient Object Detection" [67]. As a result, our decoder consists of two powerful branches. The first branch develops an initial coarse map. This technique of a two-branches decoder is adopted by many other models like [13], [44], [11], [9], [29], [31], [5] and [70].

Unlike all previous researchers that use either the coarse map or some other enhancement modules to refine the features involved in the second branch, we use both the coarse map and some key features from the first branch into the second branch, as shown in Fig. 3.1. As a result, the second branch gradually refines some of its features using some specific key vectors from the first branch, mainly because of the information

these vectors hold. The pattern of mixing these key features acts as a connection between the branches and consequently they cannot easily be distinguished.

The decoder is built upon the encoder that produces $\{X_k\}_{k=1}^{3}$ features extracted from ResNet50 encoder and $\{C_k\}_{k=1}^{4}$ features extracted from SegFormer encoder.

3.1.3.1 Coarse Map

Our decoder's first branch consists of a series of computational calculations with the aim to output a premature prediction that detects the object and acknowledges its semantic global representation. To capture this global context we proceed by manipulating the benefits of Transformers technology. Towards this target, our first goal is to compute the coarse map using only X_2, C_2, C_3 and C_4 features. Initially, we fuse X_2 with C_3 to produce an intermediate feature called MF(Mixed Features) as follows:

$$MF = BConv(Concat(BConv(Inter(C_3)) \odot X_2),\, X_2), \tag{3.1}$$

where BConv$(\cdot)$ denotes a convolutional operation followed by a batch normalization and finally by a ReLU activation function, Concat$(\cdot)$ represents the concatenation along the channel dimension, Inter$(\cdot)$ is the interpolation of the input to the given spatial size and finally $\odot$ expresses the Hadamard product.

As mentioned in [11], we multiply elements to decrease their semantic gap before concatenating them. Thus, the fusion operations of our decoder mainly consist of picking the appropriate features and forwarding them to such sequential layers.

Afterwards, our decoder fuses features C_3, C_4 and MF to produce an intermediate product C_4C_3 , as follows:

$$C_{4UpX2} = BConv(Inter(Bconv(C_4))) \tag{3.2}$$

$$C_3' = BConv(Inter(MF)) \tag{3.3}$$

$$C_4C_3 = BConv(Inter(BConv(Concat(C_{4UpX2} \odot C_3',\, C_{4UpX2})))), \tag{3.4}$$

where C_{4UpX2} occurs from C_4 after a BConv$(\cdot)$ operation, followed by upsampling X 2 for shape matching and another BConv$(\cdot)$ operation. For shape matching we also downsample X 2 the MF feature and pass it through a BConv$(\cdot)$ operation to produce C_3'.

We finally compute the coarse map as follows:

$$C_{4UpX4} = BConv(Inter(Bconv(C_4))) \tag{3.5}$$

$$C_2' = BConv(C_2)) \tag{3.6}$$

$$C_4C_3C_2 = C_2' \odot C_{4UpX4} \odot C_4C_3 \tag{3.7}$$

$$C_4C_3C_2MF = Bconv(Concat(C_4C_3C_2,\, MF)) \tag{3.8}$$

$$Coarse\ Map = Inter(Pred(C_4C_3C_2MF)), \tag{3.9}$$

where C_{4UpX4} comes from C_4 after a BConv$(\cdot)$ operation, followed by upsampling X 4 for shape matching and another BConv$(\cdot)$ operation. $C_4C_3C_2$ is the output after multiplying C_2, that passed through BConv$(\cdot)$ operation for channel matching, with both C_{4UpX4} and the previously computed element C_4C_3. The coarse map concatenates $C_4C_3C_2$ with the previously computed MF element. Intuitively we argue that MF is a powerful feature since it contains information from two features, ResNet50 X_2 and SegFormer C_3. As a result, we choose to increase its contribution to the coarse map with this concatenation. Afterwards, we forward the concatenated output into a BConv$(\cdot)$ layer for channel restoration to produce $C_4C_3C_2$MF. A Pred$(\cdot)$ layer adds another convolution to shift the coarse outcome to one channel (gray-scale for binary segmentation). The process ends by upsampling X 8 to achieve same spatial dimensions between the ground truth and the coarse map.

3.1.3.2 Refined Map

As mentioned by [66], deep features from Transformers also include rich semantic information and shallower features contain important spatial information. Based also by similar thoughts from [28], [19], [17] that concern CNN's, we proceed with a sequence of actions finalizing our decoder's second branch.

In the first place, we perform a specific combination between low level features from ResNet50 X_1, X_2 with the lowest feature from SegFormer C_1. Fusing these three elements outputs the product FB (Fused Bottom), as follows:

$$X_1X_2 = BConv(Concat(X_1,\, X_2)) \tag{3.10}$$

$$C_1X_1 = BConv(C_1 \odot X_1) \tag{3.11}$$

$$C_1X_1X_2 = BConv(Inter(BConv(Concat(C_1X_1 \odot X_1X_2,\, X_1X_2)))) \tag{3.12}$$

$$FB = BConv(Concat(MF \odot C_1X_1X_2,\, MF)), \tag{3.13}$$

where it is obvious that once again we are using the enriched intermediate element MF to influence the fusion of the lowest features from our encoders.

There exists one more powerful feature in RACOD-Net architecture. C_4C_3, as calculated from equation (3.4), is a hybrid outcome from successfully fusing C_4, C_3 and X_2. Although it contains some local properties from X_2, it is very sensitive in semantics since C_4 has a higher presence. This feature is included during this refinement stage on purpose to inject even more conceptual knowledge. We finally produce the refined map as follows:

$$F1 = C_4C_3C_2MF + FB \tag{3.14}$$

$$F2 = ChannelAttn(F1) \odot F1 \tag{3.15}$$

$$F3 = BConv(Concat(C_4C_3,\, FB)) \tag{3.16}$$

$$F4 = ChannelAttn(F3) \odot F2 \tag{3.17}$$

$$RefinedMap = Inter(Pred(F4)), \tag{3.18}$$

where the usage of C_4C_3 is obvious during the concatenation with the product FB . It is worth mentioning that similar to the skip connections deployed in ResNet architecture [18] we perform element wise addition between $C_4C_3C_2$MF and FB. The main reason is the need to pass high-level semantic information unchanged to the latter layers of our neural network. ChannelAttn$(\cdot)$ represents a new layer of channel attention that requires further analysis.

One major factor in our refinement process is the Channel-Attention layer based by a convolutional block attention module, as mentioned in [65]. This attention module, originally comes with two sub-modules, as shown in Fig. 3.3.

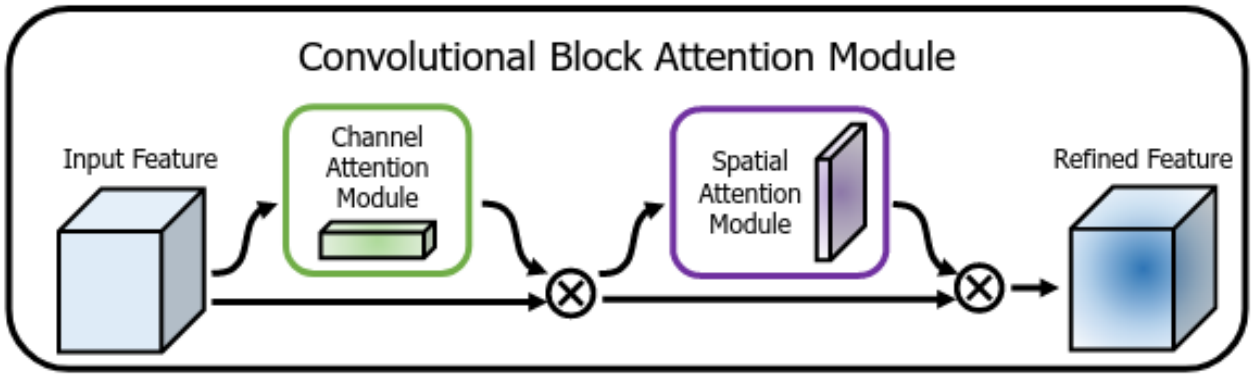


Figure 3.3: CBAM: Convolutional Block Attention Module [65]

However, unlike [65] which places these sub-modules during the ResNet50 encoder computations we placed only the channel attention sub-module in our decoder to refine the outcome. Since the performance of

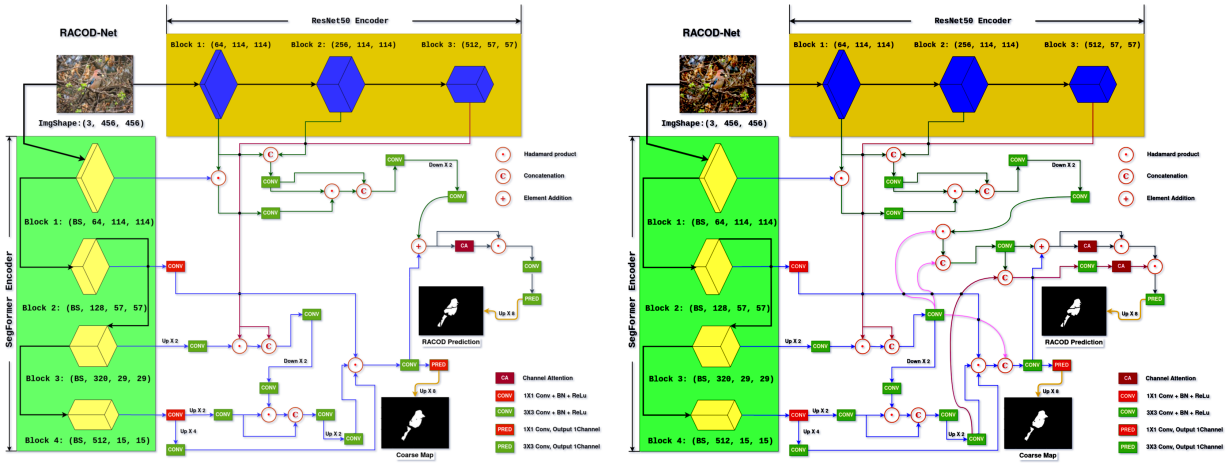
RACOD-Net increases as the channel dimension increases, it is vital to filter theses channels. Each channel in the decoder's computations is considered as a possible feature and as a result through this channel attention we are able to focus only on meaningful channels. Average and max pooling operations are executed among others to learn the extend of an object and its distinctive properties respectively. The channel attention output is computed, as follows:

$$ChannelAttn(F) = \sigma(Conv(ReLU(Conv(AvgPool(F)) + \sigma(Conv(ReLU(Conv(MaxPool(F)), \quad (3.19)$$

where σ denotes the sigmoid function, AvgPool applies average pooling and MaxPool applies max pooling.

3.1.4 Further Explanation

As mentioned earlier, our proposed architecture fuses in a unique matter the shallower layers produced from our encoders. This technique finally outputs an intermediate product termed FB (Fused Bottom), as computed from equation (3.13). Such a low-level information tensor has to be manipulated appropriately before further fusion occurs with our high semantic tensors, calculated by the first branch of our decoder. As shown in Fig. 3.4, initially there existed a naive implementation of RACOD-Net.



Naive Implementation RACOD-Net (Ours)

Figure 3.4: Visual comparison among our naive implementation and our renovated published architecture.

Our naive implementation was also based on our initial motivation. To capture both local and global context it seemed pretty straightforward to proceed with a simple addition among the low-level and the high-level information tensors, as shown in the left part of Fig. 3.4. However, this plain addition would result in a mediocre final prediction that showcased minor differences from the coarse prediction. These two tensors represent different philosophies and technologies, even though we have leaked some local properties into the global context and the opposite, by fusing X_2 with SegFormer and C_1 with ResNet50. Therefore, performing such an addition is not advisable, since these two predictions have very few common similarities and their respective weights consist of extremely divergent values.

The fundamental motivation was altered and enhanced, based by the original ResNet architecture where the skip connections add the weights from one layer to another. These sequential weights share common conceptual information since they are the outcome of adjacent layers of convolutions. Such an addition would not be effective if these weights were generated from long-distanced layers. Taking the previous analysis into account, it was mandatory to decrease the semantic gap between the decoder's predictions before any further action took place.

Towards this target, two specific hybrid features are highlighted to lessen the previously stated differences. The intermediate product MF, as calculated from equation (3.1) is the first candidate for this task. SegFormer encoder performs multiple self-attention repetitions for every generated feature. More specific, C_3 requires

27 repetitions of self-attention, as implemented by RACOD-Net. Later, C_3 is mixed with X_2 producing the intermediate product termed MF. Since, MF contains both global and local properties, when it gets multiplied and concatenated with the shallower fusion of features C_1, X_1 and X_2, it contributes the first step to slowly decrease the semantic distance among the two branches. Afterwards, the previously mentioned addition takes place and passes through a channel attention module producing feature F2, according to equations (3.14) and (3.15). Still, feature F2 is not capable enough to be forwarded as our refined accurate prediction. From visual inspection of our results, training our model with F2 as our final outcome surpasses by a large margin our naive implementation. The achievement of more precise and robust binary maps indicates that this is the right path, when fusing different technologies. However, when inspecting visually our segmentations we discovered that, despite achieving detailed boundaries and suppressed noise, there existed certain predictions that falsely classified non-camouflaged objects as camouflaged.

Following the same concept that feature MF taught us, we considered whether we could further enhance our output maps and reduce their inaccuracies. Such mistakes require more injection of semantic knowledge into feature F2. C_4C_3, as calculated from equation (3.4), is the second candidate for the completion of our final architecture. C_4C_3 is the last and deeper feature from SegFormer drawing information from the whole image. It captures distant semantic relevances among crucial parts of the given image, since it contains all the previous self-attention repetitions that SegFormer performs. F2 lacks from excessive perceptual information and C_4C_3 can handle this requirement. However, once again before multiplying these elements their semantic distance has to decreased. As shown by equations (3.16), (3.17) and Fig. 3.1, we proceed by convolving the concatenation of C_4C_3 and FB and later using a channel attention module. Finally, the produced feature F4 can be multiplied with F2 to diminish any remaining gaps developing the final enriched tensor. This tensor tracks long-range dependencies, aggregates global context and efficiently holds the necessary local convolutional properties.

There exists one major difference among our implementations. Our naive deployment avoids the production of the intermediate product termed $C_4C_3C_2$MF, as calculated from equation (3.8). Since, feature MF influences the second branch of our decoder, as described earlier, we choose in our final implementation to increase its contribution in the first branch as well. Consequently, MF is passed down to a concatenation layer, according to equation (3.8) generating a new aggregated feature. These two branches are tailored in a specific manner to attend each other. Not only the second branch is semantically close to the first branch but also through the previous concatenation the first branch lean towards the second.

This entire motivation of fusing different encoders, a CNN and a Transformer one, requires a two-way interaction between them. Following such a path and selecting these key features to perform all the previous aggregations came intuitively, through the deeper understanding of our available encoder technologies.

3.1.5 Loss Function

The total loss function during training RACOD-Net can be formulated as follows:

$$L = L_{cm} \quad + \quad L_{rm},$$

where L_{cm} is the loss function calculated between the intermediate coarse map P1 and GT (ground truth) and L_{rm} is calculated between the final refined segmentation map P2 and GT.

L_{cm} can be written as:

$$L_{cm} = L^w_{IoU}(P1, GT) \quad + \quad L^w_{BCE}(P1, GT)$$

L_{rm} can be written as:

$$L_{rm} = L^w_{IoU}(P2, GT) \quad + \quad L^w_{BCE}(P2, GT),$$

where $L^w_{IoU}(\cdot)$ and $L^w_{BCE}(\cdot)$ are the weighted intersection over union (IoU) and weighted binary cross entropy loss (BCE) respectively. In Salient Object Detection both BCE and IoU functions treat all pixels equally. However, the weighted versions of these functions based by [5], [64] set to each pixel a weight α. Hard pixels are assigned a high value of α and simple pixels are assigned a smaller value of α. We could argue that hard pixels are located in areas of the image with cluttered or elongated objects whereas simple pixels are found

in smoother areas of the image.

Unlike, [5], [64] where the weighted factor is computed as follows:

$$a = 1 \quad + \quad 5\,|AvgPool(GT) \quad - \quad GT)|,$$

RACOD-Net computes the weighted factor differently, as shown below:

$$a = \sigma(|P_i \quad - \quad GT|),$$

where σ denotes the sigmoid function and P_i with $i = 1, 2$ represents either the coarse map or the refined map. It is obvious, that if the prediction is far away from the ground-truth the loss function will be more penalized compared to the scenario where the prediction is more close to the ground-truth. To further explain this aspect the sigmoid function is calculated as follows:

$$\sigma = 1/(1 \quad + \quad e^{-x}),$$

where $x = |P_i - GT|$ with $i = 1, 2$. Since x is either zero or a positive number the output of the sigmoid function ranges between 0.5 and 1. As a result, we reward our loss function by multiplying with a factor close to 0.5 when our prediction is accurate and we penalize our loss function by multiplying with a factor close to 1 when our prediction fails by a large margin.

3.2 Datasets

We evaluate our model on the following public datasetes:

- CHAMELEON [56]
- CAMO [27]
- COD10K [11]
- NC4K [34]

Specifically, CHAMELEON contains 76 images which were taken by independent photographers who marked these as good examples of camouflaged animals. The images from CHAMELEON were collected from Google image search using the keyword "camouflaged animal". CAMO dataset consists of 1250 images, where each image includes at least one camouflage object from a variety of challenging scenarios. COD10K originally comes with 10,000 images. However, only 5,066 images contain camouflage objects covering 78 camouflaged object categories. Finally, NC4K is the largest testing dataset containing 4,121 images for effective model evaluation. NC4K was initially manufactured to evaluate the generalization ability of existing models. All datasets come with pixel-wise ground-truths manually annotated to each image.

Following, previous studies like [11], [70], [31] we use 1,000 images from CAMO and 3,040 images from COD10K for training our model. After training we evaluate our model in the entire CHAMELEON and NC4K datasets. Our evaluation set of images is completed by adding the rest 250 and 2,026 images from CAMO and COD10K respectively.

The images for testing and training our model from both CAMO and COD10K are predefined and not randomly selected. As a result, when comparing RACOD-Net with other models we are using the same datasets with the same data partitions.

3.3 Evaluation Methods

Following previous studies like [70], [25], [11], [77], [72] we evaluate our model using four evaluation metrics including Structure-measure (S_m) [7], weighted F-measure (F_m^w) [35], adaptive E-measure (αE) [8] and Mean Absolute Error (MAE) [42].

Mean Absolute Error is the mean value of the absolute difference between the prediction of a computer

vision model and the ground truth. Although, this metric provides an estimation of the dissimilarity between two vectors it fails to determine where the error takes place in the image. E-measure designed originally for binary map evaluation takes into account pixel-wise matching and global statistics, which are related to human visual perception. Weighted F-measure represents an exhaustive metric that combines both recall and precision. Additionally, camouflage object segmentation requires a method that can compare region-aware and object-aware structural similarity between predictions and ground truth. S-measure deals effectively with this requirement.

3.4 Implementation Details

RACOD-Net is implemented using the PyTorch library [41]. Our backbone ResNet50 encoder is initialized with pretrained weights on ImageNet-1K_V2 currently available from torchvision.models subpackage. Our SegFormer encoder is also initialized with pretrained weights on ImageNet-1K currently available from the authors of SegFormer. To update our network parameters during training, Adam optimizer [26] is deployed with an initial learning set to 2e-5. To help our model generalize ever better, as stated in [33], we set weight decay in Adam optimizer equal to the initial learning rate.

Moreover, our backbone SegFormer encoder comes out with several proposals over its architecture hyper-parameters. These proposals mainly differ in the channel dimension and the amount of repetitions that have to be performed in order to execute an entire block, as shown in Fig.3.2. We choose to skip the heavy-weight proposal from the authors of SegFormer named Mit-b5 and adopt the next best version Mit-b4 that has the same channel dimension but less repetitions in the mentioned blocks. Mit-b4 delivers great results and since it is a bit more light-weight in parameters comparing to Mit-b5 it reduces training time. Additionally, all predefined versions of SegFormer randomly drops entire blocks during training using a stochastic depth method introduced by [21]. Although, this method decreases substantially training time and improves the evaluation results, it is applied through SegFormer on enormous datasets. On the contrary, our training dataset consists of 4,040 images and even though the model architecture is very complicated we have discovered from our experiments that avoiding this step doesn't affect the training period and simultaneously enhances slightly our predictions. We argue that it is not advised to randomly drop blocks of our neural network when training is such a small dataset.

Our competitive results relies also on the learning rate strategy applied in our optimizer during training. Unlike [70] that deploys a cosine learning rate strategy , [72] that uses a linear warm-up and linear decay strategy or [77] where the learning rate is manually changed at 20 and 40 epochs we use a novel hybrid version of cosine learning rate with a warm restart. The terminology warm restart allows the model to reset the learning rate to the initial one. This divergence of the learning rate introduced by [32] helps our model to escape from a potential local minima and improve the convergence into a global minimum. However, after observing our loss during training we noticed that after epoch 14 our model is slowly staring to reach convergence. From this point in training we initialize the cosine learning rate strategy with warm restart, as displayed in Fig. 3.5.

We deployed during training an automated mixed precision strategy from the PyTorch library [41], avoiding the Apex Nvidia PyTorch extension. In the course of training any model CUDA operations, executed by GPU, run in a specific data type (dtype) object chosen by autocast to improve performance while maintaining accuracy. Essentially we obtain the speed and memory usage benefits of lower precision data types, in our case float16 instead of float32, while preserving convergence behavior. Afterwards, gradient scaling is applied to prevent gradients with small magnitudes, of our model, from flushing to zero. This gradient scaling enlarges the magnitude of gradients just before backpropagation starts.

During training we set the batch size equal to 6, the images are resized to 456×456 resolution and are randomly splitted into those batches without any post-processing procedures. The channel dimension of our model is set to 768. The output predictions of our model are resized to match the original binary ground truths during evaluation. Following [25], [31], [11], [5], [79] after reshaping our prediction we forward it through a sigmoid function and later we normalize it. Since some predictions might contain very small camouflage objects resulting to an almost black segmentation output we add a very small factor equal to 1e-8

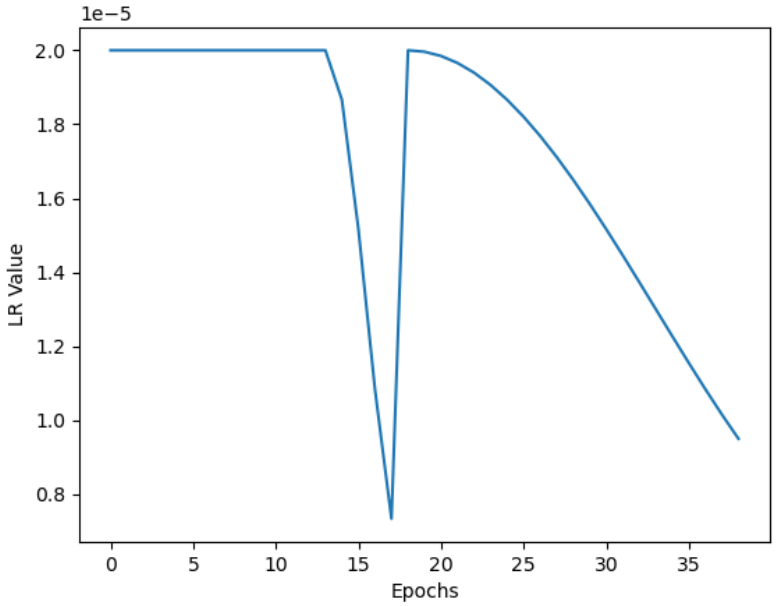


Figure 3.5: RACOD-Net Learning Rate Strategy

to avoid division by zero during normalization.

We train our model end-to-end for 39 epochs with a cloud GPU P100 16Gb for 11 hours. Alternatively, the same training process costs only 6 hours and 30 minutes when using a Zotac GeForce RTX 3060 Twin Edge 12Gb. When training with batch size of 6 it is required a GPU with at least 12Gb memory. As the batch size increases it is mandatory to obtain a GPU with at least 16Gb of memory. The inference time is approximately 0.4 seconds for an input image of size 456×456.

Chapter 4

Experiments

4.1 Results Over Camouflaged Datasets

We compare RACOD-Net with several state-of-the-art camouflaged object detection studies that deploy either a CNN or a Transformer based architecture. As shown in Tab. 1, RACOD-Net surpasses almost all methods in several evaluation metrics in almost all datasets. Even when RACOD-Net's predictions come second, they are still very close from reaching the top. For fair comparison, all the predictions are evaluated using the same evaluation metrics and the same evaluation code. Additionally, all the camouflaged maps prediction scores are provided either by the authors or generated by retraining the models with the provided open source codes.

Through quantitative and qualitative analysis over previous studies we aim to analyze our model regarding its learning ability and its generalizability over camouflaged objects. A total of 11 CNN-based comparisons are displayed in Tab. 1, including PraNet [12], SINet [11], SLSR [34], MGL-R [73], PFNet [36], C^2FNet [57], SINetV2 [10], DGNet [24], SegMaR [25], ZoomNet [72] and FDNet [77] and 5 Transformer-based comparisons, including UGTR [77], TPRNet [74], DTINet [31], CamoFormer-S [70] based in Swin-Transformer encoder [30] and CamoFormer-P [70] adopting a pyramid vision transformer encoder (PVTv2 [62]).

Method	NC4K				COD10K-Test				CAMO-Test				CHAMELEON			
	4,121 Images				2,026 Images				250 Images				76 Images			
	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$
PraNet$_{2020}$ [12]	.822	.871	.724	.059	.789	.839	.629	.045	.769	.833	.663	.094	.860	.898	.763	.044
SINet$_{2020}$ [11]	.808	.883	.723	.058	.776	.867	.631	.043	.745	.825	.644	.092	.872	.938	.806	.034
SLSR$_{2021}$ [34]	.840	.902	.766	.048	.804	.882	.673	.037	.787	.855	.696	.080	.890	.936	.822	.030
MGL-R$_{2021}$ [73]	.833	.893	.739	.053	.814	.865	.666	.035	.782	.847	.695	.085	.893	.923	.812	.031
PFNet$_{2021}$ [36]	.829	.892	.745	.053	.800	.868	.660	.040	.782	.852	.695	.085	.882	.942	.810	.033
C^2FNet$_{2021}$ [57]	.838	.898	.762	.049	.813	.886	.686	.036	.796	.864	.719	.080	.888	.932	.828	.032
UGTR$_{2021}$ [77]	.839	.886	.746	.052	.817	.850	.666	.036	.784	.859	.794	.086	.888	.921	.794	.031
SINetV2$_{2022}$ [10]	.847	.898	.770	.048	.815	.863	.680	.037	.820	.875	.743	.070	.888	.930	.816	.030
DGNet$_{2022}$ [24]	.857	.907	.784	.042	.822	.877	.693	.033	.839	.901	.769	.057	.890	.934	.816	.029
SegMaR$_{2022}$ [25]	.841	.905	.781	.046	.833	.895	.724	.033	.815	.872	.742	.071	.897	.950	.835	.027
ZoomNet$_{2022}$ [72]	.853	.907	.784	.043	.838	.893	.729	.029	.820	.883	.752	.066	**.902**	.952	**.845**	**.023**
FDNet$_{2022}$ [77]	.834	.895	.750	.052	.837	.897	.731	.030	.844	.903	.778	.062	.894	.948	.819	.030
TPRNet$_{2022}$ [74]	.854	.903	.790	.047	.829	.892	.725	.034	.814	.870	.781	.076	.891	.930	.816	.031
DTINet$_{2022}$ [31]	.863	**.915**	.792	**.041**	.824	.893	.695	.034	.857	.912	.796	.050	.883	.928	.813	.033
CamoFormer-S$_{2022}$ [70]	**.888**	**.941**	**.840**	**.031**	**.862**	**.932**	**.772**	**.024**	**.876**	**.935**	**.832**	**.043**	.891	**.953**	.829	.026
CamoFormer-P$_{2022}$ [70]	**.892**	**.941**	**.847**	**.030**	**.869**	**.931**	**.786**	**.023**	**.872**	**.931**	**.831**	**.046**	**.910**	**.970**	**.865**	**.022**
RACOD-Net (Ours)	**.889**	**.939**	**.855**	**.031**	**.872**	**.942**	**.804**	**.022**	**.868**	**.928**	**.835**	**.047**	**.917**	**.971**	**.887**	**.021**

Table 1: Quantitative results on public datasets. Results from previous studies are verified by [70], [31], [25], [36] and [24]. RACOD-Net outperforms state-of-the-art models in several scenarios. Red, Green, and Blue indicate the best, second best and third best performance. '↑/↓' denotes that the higher/lower the score, the better.

4.2 Discussion

COD10K is the most challenging dataset for camouflaged objects. The comparison in Tab. 1, demonstrates that our model delivers the best results over COD10K setting new state-of-the-art records. This high-end performance in such a compelling dataset is built upon details. These details derive from paying attention to both local information and global semantics. This suggests that such fusion verifies the effectiveness of our proposed model and justifies our original motivation and effort.

Our proposed model achieves the best performance in CHAMELEON across all metrics, by a small margin. This performance gain over previous studies proves the value of fusing properly traditional convolutional networks with Transformer networks.

Over CAMO and NC4K dataset our model surpasses several previous public well-trained models. Although, performance over weighted F-measure (F_m^w) is outstanding and sets a new record, our prediction maps for the rest evaluation metrics captures either the second or the third best performance.

From a total of 16 evalutions, 4 for each dataset, RACOD-Net demonstrates its value by providing the leading results in 62.5% of all cases. We argue that our final segmentation results are very close to the ground-truth annotations, by successfully segmenting not only large camouflaged objects but also small ones. From Fig. 1.1 and various other results we observed that our method successfully segments the position of camouflage objects with accurate and precise boundaries over several challenging scenes, such as multiple and low-contrast objects. Even when some camouflaged objects are divided into separate parts because of

the interference with other non-camouflaged objects RACOD-Net is still capable of detecting and segmenting the expected target.

4.3 Ablation Studies

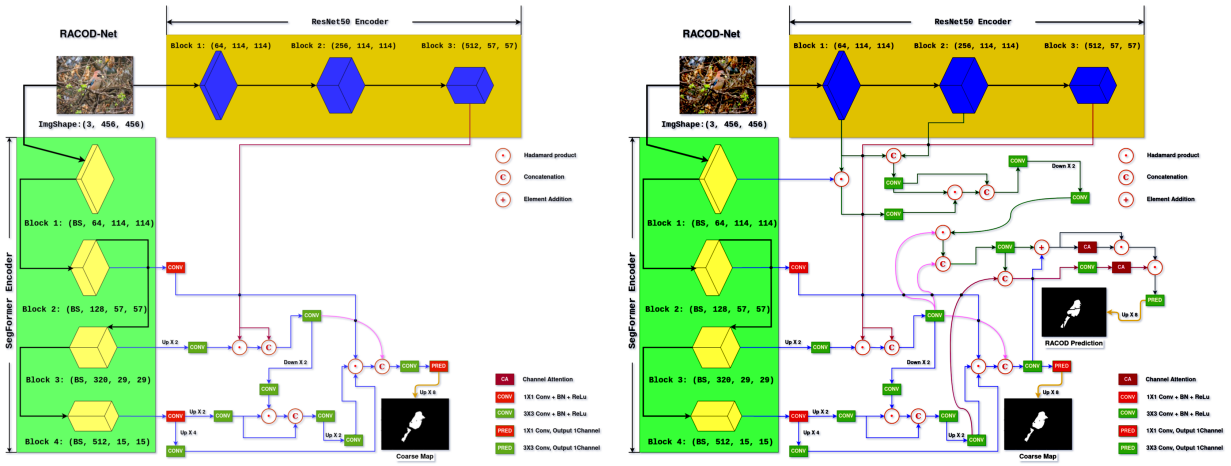
We conduct ablation studies to strengthen our decisions over particular components of RACOD-Net's architecture. The main motivation of providing accurate and precise segmentation predictions relies upon significant key aspects tailored specifically for this task. Thus, it is mandatory to demonstrate the effectiveness of these components. It is worth mentioning, before proceeding to more detailed informations that the produced results retain some randomness over different training cycles. Even so, it is safe to state that our submitted architecture distributes the finest predictions across all the following benchmarks. Quantitative experimental results of the ablation studies are shown in Table 2.

Settings	NC4K				COD10K-Test				CAMO-Test				CHAMELEON			
	4,121 Images				2,026 Images				250 Images				76 Images			
	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$	$S_m\uparrow$	$\alpha E\uparrow$	$F_m^w\uparrow$	$M\downarrow$
Only Coarse Map	.886	.931	.834	.035	.870	.918	.778	.025	.865	.917	.812	.052	.913	.949	.857	.026
Naive Implementation	.872	.928	.837	.037	.860	.938	.792	.024	.847	.915	.815	.055	.902	.957	.872	.025
C_D=256	.884	.936	.845	.033	.868	.942	.796	.022	.861	.919	.823	.050	.910	.960	.870	.024
C_D=128	.886	.936	.842	.033	**.872**	.937	.794	.023	.864	.925	.825	.049	.915	.960	.874	.022
C_D=64	.887	.934	.837	.034	.868	.922	.779	.024	.866	.922	.820	.050	.908	.944	.855	.027
Image Size: 256×256	.848	.916	.800	.044	.834	.928	.751	.029	.801	.870	.746	.071	.850	.921	.791	.038
RACOD-Net (Ours)	**.889**	**.939**	**.855**	**.031**	**.872**	**.942**	**.804**	**.022**	**.868**	**.928**	**.835**	**.047**	**.917**	**.971**	**.887**	**.021**

Table 2: Quantitative results of ablation studies. The best scores are highlighted in bold. '↑/↓' denotes that the higher/lower the score, the better.

4.3.1 Effectiveness of the Refined Map

As already mentioned, the process of predicting with accuracy the boundaries of a camouflage object requires from our model to refine the original predicted coarse map. The refinement process passes through many stages and fuses several features with an organised manner. In order to validate the existence of the refinement process we remove this entire block and evaluate only the coarse map. We re-train our model from the start by placing in our decoder only the first branch that calculates the coarse map. We also alter our loss function to expect only one prediction instead of two, since now we have available only one branch. Visually our altered architecture is displayed in Fig. 4.1. The computation cost is substantially decreased, since the entire second branch is missing. Additionally, several features from our encoders are extracted but are not exploited for further aggregations, rendering our backbone impractical.



Coarse Map RACOD-Net (Ours)

Figure 4.1: Visual comparison of architectures with or without the refinement map.

As shown in Tab. 2, the performance of our model decreases among all datasets. The differences may appear insignificant but are crucial since the noise of the refined map is suppressed and its structure is more clear and robust. Not only boundaries are more precise but the main body of the object appears more compact, by detecting and eliminating internal empty spots. From visual inspection over some samples in Fig. 4.2, the differences with the refinement map or without are obvious, setting our second branch a mandatory element.

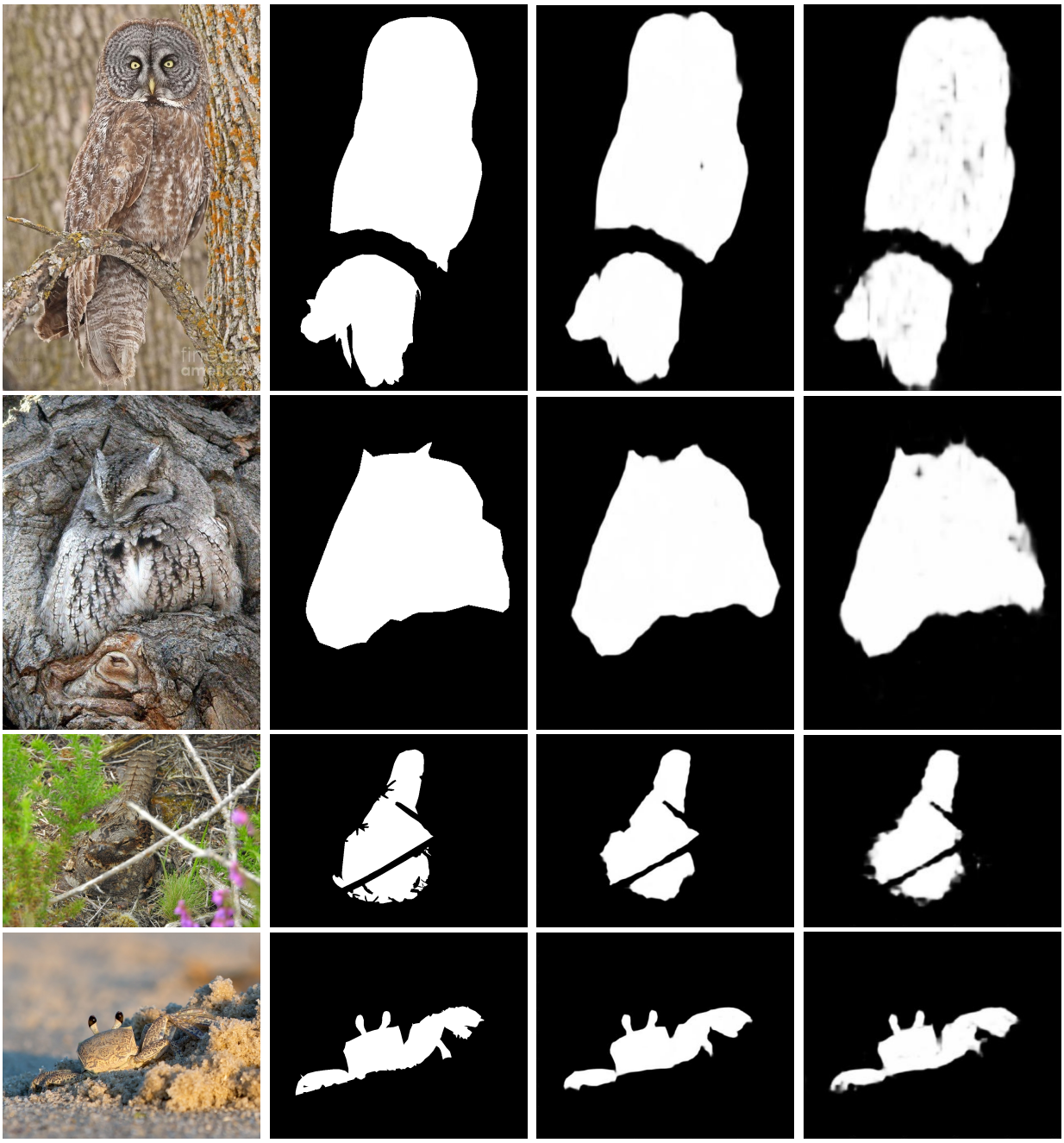


Image GT RACOD-Net (Ours) Only Coarse Map

Figure 4.2: Ablation study with the refined map or without.

4.3.2 Effectiveness of Naive Implementation

As pointed out during our architecture analysis, initially there existed a naive implementation of RACOD-Net. The two decoder branches would interact with each other through a simple addition of the produced tensors, as shown in Fig. 3.4. We aim, through, this experiment to strengthen the analysis of section 3.1.4, where we claimed that the primitive results of our naive implementation were mediocre and barely tolerable. Before, any visual comparisons take place we must pinpoint the results from Tab. 2. Our naive implementation achieves almost the same results with our first experiment, where we used only the coarse map of our decoder. From these evaluation comparisons it is easily noticed that the final refined prediction from

the premature implementation is as good as the coarse map prediction of our final published architecture. Through this observation process we have verified that when fusing different features from a CNN and a Transformer encoder a simple addition of the produced tensors is not enough. The visual differences in Fig. 4.3 fortify our decisions and our intuition, regarding the selection of features to unify our decoder's two branches.

Figure 4.3: Ablation study comparing our original implementation with the premature naive one.

All camouflaged objects in Fig. 4.3 are very challenging and their boundaries are difficult to distinguish. RACOD-Net delivers great predictions compared to the naive implementation that behaves similar to the coarse prediction, failing to detect details and large portion of the object's body.

4.3.3 Effectiveness of Channel Dimension

We further analyze the influence of channel dimension over performance. We observe that the overall behaviour of our model increases among all evaluation metrics as the channel dimension becomes greater. However, the parameters and the computation cost increases as well. Our original architecture with 768 channels consists of almost 132M parameters. Setting channel dimension to 256 or 128 we drop either to 93M or 88M parameters. Decreasing the channels we still manage to deliver a competitive performance with exceptional results and deliver a real-time experience, in case of a mobile deployment. Still we have to mention, that lowering the channels translates in our model picking less features from a given object. From Fig. 4.4, it is noticeable that less channels lead to objects with fewer details. For instance, from the images of the second row the details of the camouflaged lizard fade as we get closer to 64 channels.

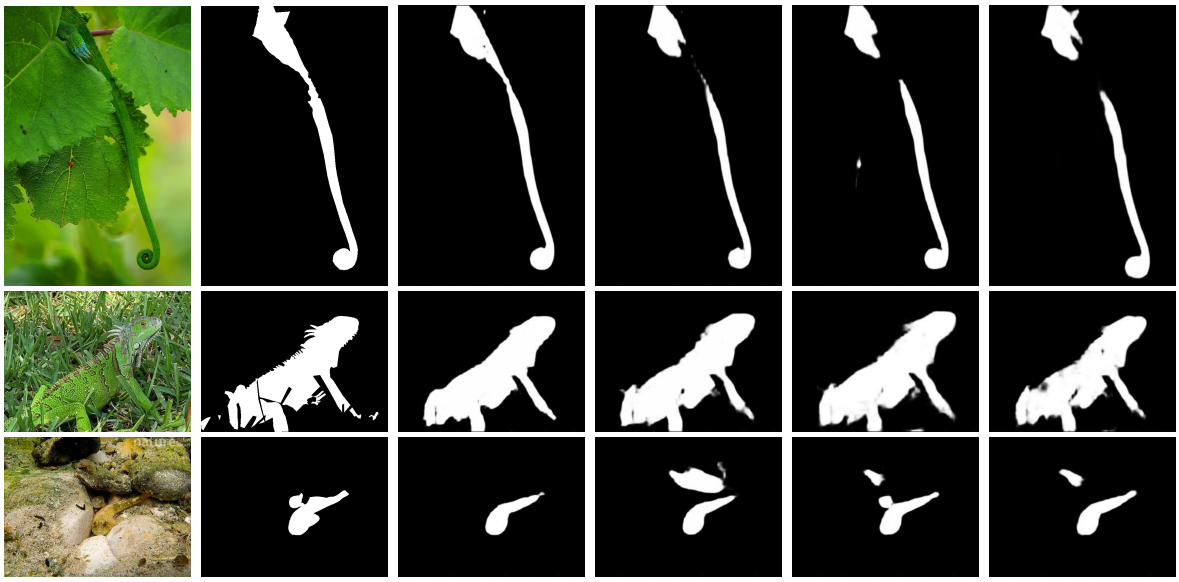


Figure 4.4: Ablation study regarding the channel dimension.

4.3.4 Effectiveness of Image Size

We considered decreasing the image size from 456×456 to 256×256, like [31] that utilized a dual SegFormer encoder using a similar batch size like ours. COD10K is the dominant dataset during training and the majority of the provided images have an approximate resolution around 1024×768. Even though it is preferred to insert images with high spatial dimensions to provide better quality input to the backbone encoders at the same time it is resource consuming. We find that setting resolution higher than 465×456 leads to performance saturation and requires more GPU memory. On the other hand, setting resolution to 256×256 decreases the execution time dramatically but generates a major performance degradation, as shown in Fig 4.5. The main core of the object is still detected but the details regarding its boundary and structure appear to be less consistent and accurate.

Figure 4.5: Ablation study regarding the input image size.

4.4 Results Over Polyp Datasets

We compare RACOD-Net with several polyp segmentation studies. As shown in Tab. 3 and Tab. 4, RACOD-Net architecture originally designed for camouflaged object detection and segmentation achieves great results in polyp segmentation tasks as well. For fair comparison, all the predictions are evaluated using the same evaluation metrics and the same evaluation code. Additionally, all the binary prediction scores are provided either by the authors or generated by retraining the models with the provided open source codes.

Polyp segmentation tasks are similar to camouflaged object detection tasks. Both assignments belong to the general semantic segmentation computer vision field. Pixels belonging to a segmented polyp over some prediction are assigned with the value 1, otherwise they are assigned with the value 0. Additionally, the color and texture of polyps blend with the surrounding healthy tissues, thus gaining some camouflaged properties. Evaluating our model over polyp segmentation datasets seems a great step towards discovering whether or not our architecture has the potential of being deployed in several other fields of computer vision.

We adopt five challenging public datasets, including Kvasir-SEG [23], ETIS [53], CVC-ClinicDB [2], CVC-ColonDB [59] and CVC300 [61]. Following the same setup like PraNet [12] and Polyp-PVT [5] we use the same training and evaluation datasets. More specific our training dataset consists of a total of 1,450 images from Kvasir-SEG and CVC-ClinicDB. Our evaluation dataset consists of 100 images from Kvasir-SEG, 62

images from CVC-ClinicDB, 196 images from ETIS, 380 images from CVC-ColonDB and 60 images from CVC300 dataset. Regarding the evaluation metrics, instead of adaptive E-measure we utilize the mean E-measure. All the rest evaluation criteria remain intact.

However we altered three minor hyper-parameters of our model before proceeding with training. The batch size increased from 6 to 8, the input image size is reshaped from 456×456 to 352×352 and the initial learning rate is set to 4e-5 from 2e-5. Since our training dataset is even smaller than the previous camouflaged dataset we considered increasing the batch size and the initial learning rate. However, from visual inspection of the provided images their average spatial dimension is much smaller than the average spatial dimensions of the camouflaged images. Thus, it appears preferable to adopt a smaller input image size in our encoders.

A total of 9 comparisons are displayed in Tab. 3 and Tab. 4, including , U-Net (MICCAI'15) [51], UNet++ (DLMIA'18) [78], SFA (MICCAI'19) [14], MSEG [20], DCRNet [71], ACSNet (MICCAI'20) [75], PraNet (MICCAI'20) [12], SANet (MICCAI'21) [63] and Polyp-PVT [5].

Method	Kvasir-SEG 100 Images				CVC-ClinicDB 62 Images				CVC-ColonDB 380 Images			
	$S_m\uparrow$	mE↑	$F_m^w\uparrow$	M↓	$S_m\uparrow$	mE↑	$F_m^w\uparrow$	M↓	$S_m\uparrow$	mE↑	$F_m^w\uparrow$	M↓
U-Net (MICCAI'15) [51]	.858	.881	.794	.055	.889	.913	.811	.019	.712	.696	.498	.061
UNet++ (DLMIA'18) [78]	.862	.886	.808	.048	.873	.891	.785	.022	.691	.680	.467	.064
SFA (MICCAI'19) [14]	.782	.834	.670	.075	.793	.840	.647	.042	.634	.675	.379	.094
MSEG $_{2021}$ [20]	.912	.942	.885	.028	.938	.961	.907	.007	.834	.859	.724	.038
DCRNet$_{2021}$ [71]	.911	.933	.868	.035	.933	.964	.890	.010	.821	.840	.684	.052
ACSNet (MICCAI'20) [75]	.920	.941	.882	.032	.927	.947	.873	.011	.829	.839	.697	.039
PraNet (MICCAI'20) [12]	.915	.944	.885	.030	.936	.963	.896	.009	.820	.847	.699	.043
SANet (MICCAI'21) [63]	.915	.949	.892	.028	.939	.971	.909	.012	.837	.869	.726	.043
Polyp-PVT$_{2022}$ [5]	.925	.956	.911	.023	**.949**	**.985**	**.936**	**.006**	**.865**	**.913**	**.795**	**.031**
RACOD-Net (Ours)	**.935**	**.964**	**.923**	**.020**	.938	.965	.915	.014	.855	.893	.772	.032

Table 3: Quantitative results on 3 public datasets, including Kvasir-SEG [23], CVC-ClinicDB [2] and CVC-ColonDB [59]. Results from previous studies are verified by [5]. The best scores are highlighted in bold '↑/↓' denotes that the higher/lower the score, the better.

Method	ETIS				CVC300			
	196 Images				60 Images			
	$S_m\uparrow$	mE↑	$F_m^w\uparrow$	M↓	$S_m\uparrow$	mE↑	$F_m^w\uparrow$	M↓
U-Net (MICCAI'15) [51]	.684	.643	.366	.036	.843	.847	.684	.022
UNet++ (DLMIA'18) [78]	.683	.629	.390	.035	.839	.834	.687	.018
SFA (MICCAI'19) [14]	.557	.531	.231	.109	.640	.644	.341	.065
MSEG $_{2021}$ [20]	.828	.854	.671	.015	.924	.948	.852	.009
DCRNet$_{2021}$ [71]	.736	.742	.506	.096	.921	.943	.830	.010
ACSNet (MICCAI'20) [75]	.754	.737	.530	.059	.923	.939	.825	.013
PraNet (MICCAI'20) [12]	.794	.808	.600	.031	.925	.950	.843	.010
SANet (MICCAI'21) [63]	.849	.881	.685	.015	.928	.962	.859	.008
Polyp-PVT$_{2022}$ [5]	**.871**	**.906**	**.750**	**.013**	.935	**.973**	.884	.007
RACOD-Net (Ours)	.863	.894	.727	.014	**.942**	.965	**.885**	**.006**

Table 4: Quantitative results on 2 public datasets, including ETIS [53] and CVC300 [61]. Results from previous studies are verified by [5]. The best scores are highlighted in bold '↑/↓' denotes that the higher/lower the score, the better.

RACOD-Net, despite being designed for camouflaged object detection and segmentation, delivers great results and even sets some new records over Kvasir-SEG and CVC300 datasets. Almost all our predictions are pretty close from reaching the finest results. We argue that through further experiments RACOD-Net's hyper-parameters could be fine-tuned to deliver state-of-the-art records among a larger portion of polyp datasets. Still, the overall performance is very promising and verifies the effectiveness of our model over certain computer vision assignments.

Chapter 5

Conclusion

5.1 Future Work

In this paper we proposed a novel fusion between traditional neural networks and Transformer based networks. We proved that such fusion demonstrates great results setting new records over several metrics. All modules were carefully combined and all hyper parameters were thoroughly fine tuned to get the job done. Comprehensive ablation studies also validate our contributions. However, future extensive experiments are following with the aim of developing and further evolving our original model.

There are several areas of computer vision in the field of medical science, such as segmentation of polyps and tumors. In both of these cases the color, texture and shape of polyps or tumors provide them with strong camouflaged properties. A similar task is also the segmentation of computed tomography (CT) images to distinguish normal tissues from infected ones. Such applications showed great interest during COVID-19.

We also evaluated our model on polyp segmentation datasets. The evaluation results were quite encouraging indicating that the present architecture of RACOD-Net has the potential of further generalization and deployment.

We aim to further train our model over various datasets and discover its behaviour over several tasks beyond camouflaged objects. We argue that our model, with certain adjustments and weights produced from certain datasets, could produce significant results over different computer vision tasks. Salient object detection, small object detection, video object detection and object detection in aerial images could be possible applications to extend RACOD-Net's original architecture. We acknowledge that our model suffers from high computational cost, even when the entire training dataset consists of a small amount of images. We are obliged to convert our model into a more lightweight network when dealing with tasks that involve large and compelling datasets.

We hope that RACOD-Net could act as a reference framework, stimulating more novel ideas over challenging computer vision areas. Our code, with detailed instructions, and our pretrained weights validating the results of Tab. 1, Tab. 3 and Tab. 4 have been released at: `https://github.com/mikestratakis/RACOD-Net`.

Bibliography

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015.

[2] Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Debora Gil, Cristina Rodríguez, and Fernando Vilariño. Wm-dova maps for accurate polyp highlighting in colonoscopy: Validation vs. saliency maps from physicians. *Computerized Medical Imaging and Graphics*, 43:99–111, 2015.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, abs/2005.12872, 2020.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893, 2005.

[5] Bo Dong, Wenhai Wang, Deng-Ping Fan, Jinpeng Li, Huazhu Fu, and Ling Shao. Polyp-pvt: Polyp segmentation with pyramid vision transformers, 2021.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020.

[7] Deng-Ping Fan, Ming-Ming Cheng, Yun Liu, Tao Li, and Ali Borji. Structure-measure: A new way to evaluate foreground maps, 2017.

[8] Deng-Ping Fan, Cheng Gong, Yang Cao, Bo Ren, Ming-Ming Cheng, and Ali Borji. Enhanced-alignment measure for binary foreground map evaluation. In *International Joint Conference on Artificial Intelligence*, 2018.

[9] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *CoRR*, abs/2102.10274, 2021.

[10] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE TPAMI*, 44(10):6024–6042, 2022.

[11] Deng-Ping Fan, Ge-Peng Ji, Guolei Sun, Ming-Ming Cheng, Jianbing Shen, and Ling Shao. Camouflaged object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] Deng-Ping Fan, Ge-Peng Ji, Tao Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Pranet: Parallel reverse attention network for polyp segmentation, 2020.

[13] Deng-Ping Fan, Tao Zhou, Ge-Peng Ji, Yi Zhou, Geng Chen, Huazhu Fu, Jianbing Shen, and Ling Shao. Inf-net: Automatic covid-19 lung infection segmentation from ct images. *IEEE Transactions on Medical Imaging*, 39(8):2626–2637, 2020.

[14] Yuqi Fang, Cheng Chen, Yixuan Yuan, and Kai-yu Tong. *Selective Feature Aggregation Network with Area-Boundary Constraints for Polyp Segmentation*, pages 302–310. 10 2019.

[15] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[16] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[17] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization, 2014.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[19] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, Zhuowen Tu, and Philip H. S. Torr. Deeply supervised salient object detection with short connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):815–828, apr 2019.

[20] Chien-Hsiang Huang, Hung-Yu Wu, and Youn-Long Lin. Hardnet-mseg: A simple encoder-decoder polyp segmentation neural network that achieves over 0.9 mean dice and 86 fps, 2021.

[21] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016.

[22] T Huang. Computer Vision: Evolution And Promise. 1996.

[23] Debesh Jha, Pia H. Smedsrud, Michael A. Riegler, Pål Halvorsen, Thomas de Lange, Dag Johansen, and Håvard D. Johansen. Kvasir-seg: A segmented polyp dataset, 2019.

[24] Ge-Peng Ji, Deng-Ping Fan, Yu-Cheng Chou, Dengxin Dai, Alexander Liniger, and Luc Van Gool. Deep gradient learning for efficient camouflaged object detection, 2022.

[25] Qi Jia, Shuilian Yao, Yu Liu, Xin Fan, Risheng Liu, and Zhongxuan Luo. Segment, magnify and reiterate: Detecting camouflaged objects the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4713–4722, June 2022.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

[27] Trung-Nghia Le, Tam V. Nguyen, Zhongliang Nie, Minh-Triet Tran, and Akihiro Sugimoto. Anabranch network for camouflaged object segmentation. *CoRR*, abs/2105.09451, 2021.

[28] Gayoung Lee, Yu-Wing Tai, and Junmo Kim. Deep saliency with encoded low level distance map and high level features, 2016.

[29] Chen Li and Ge Jiao. EINet: camouflaged object detection with pyramid vision transformer. *Journal of Electronic Imaging*, 31(5):053002, 2022.

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, 2021.

[31] Zhengyi Liu, Zhili Zhang, Yacheng Tan, and Wei Wu. Boosting camouflaged object detection with dual-task interactive transformer. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 140–146. IEEE, 2022.

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2017.

[34] Yunqiu Lv, Jing Zhang, Yuchao Dai, Aixuan Li, Bowen Liu, Nick Barnes, and Deng-Ping Fan. Simultaneously localize, segment and rank the camouflaged objects, 2021.

[35] Ran Margolin, Lihi Zelnik-Manor, and Ayellet Tal. How to evaluate foreground maps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2014.

[36] Haiyang Mei, Ge-Peng Ji, Ziqi Wei, Xin Yang, Xiaopeng Wei, and Deng-Ping Fan. Camouflaged object segmentation with distraction mining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[37] Sami Merilaita, Nicholas Scott-Samuel, and Innes Cuthill. How camouflage works. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372:20160341, 07 2017.

[38] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. *CoRR*, abs/1505.04366, 2015.

[39] Paraskevi Nousi, Maria Tzelepi, Nikolaos Passalis, and Anastasios Tefas. Chapter 7 - lightweight deep learning. In Alexandros Iosifidis and Anastasios Tefas, editors, *Deep Learning for Robot Perception and Cognition*, pages 131–164. Academic Press, 2022.

[40] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[42] Federico Perazzi, Philipp Krähenbühl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

[43] Tasha Price, Samuel Green, Jolyon Troscianko, Tom Tregenza, and Martin Stevens. Background matching and disruptive coloration as habitat-specific strategies for camouflage. *Scientific Reports*, 9, 05 2019.

[44] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand. Basnet: Boundary-aware salient object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7471–7481, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society.

[45] Ramin Ranjbarzadeh, Saeid Ghoushchi, Malika Bendechache, Amir Amirabadi, Mohd Ab Rahman, Soroush Saadi, Amirhossein Aghamohammadi, and Mersedeh Kooshki. Lung infection segmentation for covid-19 pneumonia based on a cascade convolutional network from ct images. *BioMed Research International*, 2021:1–16, 04 2021.

[46] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[47] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[48] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[49] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

[52] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1605.06211, 2016.

[53] Juan Silva, Aymeric Histace, Olivier Romain, Xavier Dray, and Bertrand Granado. Toward embedded detection of polyps in wce images for early diagnosis of colorectal cancer. *International journal of computer assisted radiology and surgery*, 9, 09 2013.

[54] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014.

[55] John Skelhorn and Candy Rowe. Cognition and the evolution of camouflage. *Proceedings of the Royal Society B: Biological Sciences*, 283:20152890, 02 2016.

[56] Przemysław Skurowski, Hassan Abdulameer, J Błaszczyk, Tomasz Depta, Adam Kornacki, and P Kozieł. Animal camouflage analysis: Chameleon database. unpublished manuscript, 2018.

[57] Yujia Sun, Geng Chen, Tao Zhou, Yi Zhang, and Nian Liu. Context-aware cross-level fusion network for camouflaged object detection, 2021.

[58] Yujia Sun, Shuo Wang, Chenglizhao Chen, and Tian-Zhu Xiang. Boundary-guided camouflaged object detection. In *IJCAI*, pages 1335–1341, 2022.

[59] Nima Tajbakhsh, Suryakanth R. Gurudu, and Jianming Liang. Automated polyp detection in colonoscopy videos using shape and context information. *IEEE Transactions on Medical Imaging*, 35(2):630–644, 2016.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017.

[61] David Vázquez, Jorge Bernal, F. Javier Sánchez, Gloria Fernández-Esparrach, Antonio M. López, Adriana Romero, Michal Drozdzal, and Aaron Courville. A benchmark for endoluminal scene segmentation of colonoscopy images, 2016.

[62] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *CoRR*, abs/2106.13797, 2021.

[63] Jun Wei, Yiwen Hu, Ruimao Zhang, Zhen Li, S. Kevin Zhou, and Shuguang Cui. Shallow attention network for polyp segmentation, 2021.

[64] Jun Wei, Shuhui Wang, and Qingming Huang. F3net: Fusion, feedback and focus for salient object detection, 2019.

[65] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module, 2018.

[66] Pan Wu, Limai Jiang, Zhen Hua, and Jinjiang Li. Multi-focus image fusion: Transformer and shallow feature attention matters. *Displays*, 76:102353, 2023.

[67] Zhe Wu, Li Su, and Qingming Huang. Cascaded partial decoder for fast and accurate salient object detection. *CoRR*, abs/1904.08739, 2019.

[68] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *Neural Information Processing Systems (NeurIPS)*, 2021.

[69] Fan Yang, Qiang Zhai, Xin Li, Rui Huang, Hong Cheng, and Deng-Ping Fan. Uncertainty-guided transformer reasoning for camouflaged object detection. In *IEEE International Conference on Computer Vision(ICCV)*, 2021.

[70] Bowen Yin, Xuying Zhang, Qibin Hou, Bo-Yuan Sun, Deng-Ping Fan, and Luc Van Gool. Camoformer: Masked separable attention for camouflaged object detection, 2022.

[71] Zijin Yin, Kongming Liang, Zhanyu Ma, and Jun Guo. Duplex contextual relation network for polyp segmentation, 2022.

[72] Pang Youwei, Zhao Xiaoqi, Xiang Tian-Zhu, Zhang Lihe, and Lu Huchuan. Zoom in and out: A mixed-scale triplet network for camouflaged object detection, 2022.

[73] Qiang Zhai, Xin Li, Fan Yang, Chenglizhao Chen, Hong Cheng, and Deng-Ping Fan. Mutual graph learning for camouflaged object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[74] Qiao Zhang, Yanliang Ge, Cong Zhang, and Hongbo Bi. Tprnet: camouflaged object detection via transformer-induced progressive refinement network. *The Visual Computer*, 07 2022.

[75] Ruifei Zhang, Guanbin Li, Zhen Li, Shuguang Cui, Dahong Qian, and Yizhou Yu. Adaptive context selection for polyp segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 253–262. Springer, 2020.

[76] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *CoRR*, abs/2012.15840, 2020.

[77] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4504–4513, June 2022.

[78] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested u-net architecture for medical image segmentation. *CoRR*, abs/1807.10165, 2018.

[79] Hongwei Zhu, Peng Li, Haoran Xie, Xuefeng Yan, Dong Liang, Dapeng Chen, Mingqiang Wei, and Jing Qin. I can find you! boundary-guided separated attention network for camouflaged object detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(3):3608–3616, Jun. 2022.