Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Semantic Segmentation of Coastal Images with Transformer Models

Διπλωματικη Εργασια

της

Πελαγίας Δρακοπούλου

Επιβλέπων: Στέφανος Κόλλιας
Ομότιμος Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
ΕΔΙΠ Ε.Μ.Π.

Αθήνα, Απρίλιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

Semantic Segmentation of Coastal Images with Transformer Models

Διπλωματικη Εργασια

της

Πελαγίας Δρακοπούλου

Επιβλέπων: Στέφανος Κόλλιας
Ομότιμος Καθηγητής Ε.Μ.Π.

Συνεπιβλέπουσα: Παρασκευή Τζούβελη
ΕΔΙΠ Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Απριλίου 2024.

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Στέφανος Κόλλιας	Γιώργος Στάμου	Αθανάσιος Βουλόδημος
Ομότιμος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Απρίλιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Τεχνολογιας Πληροφορικης και Υπολογιστων

.....
Πελαγία Δρακοπούλου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Οι παράκτιες περιοχές έχουν ιδιαίτερη σημασία στην ανάπτυξη διαφόρων κοινωνικοοικονομικών δραστηριοτήτων, την περιβαλλοντική βιωσιμότητα και τη διατήρηση της βιοποικιλότητας. Για τον λόγο αυτό, απαιτούνται αποτελεσματικά εργαλεία και μεθοδολογίες για την κατανόηση και την παρακολούθηση αυτών των δυναμικών οικοσυστημάτων. Η τηλεπισκόπηση, ειδικά μέσω δορυφορικών εικόνων, έχει αναδειχθεί ως μια ιδιαίτερα χρήσιμη τεχνολογία για την ανάλυση των ακτογραμμών. Οι πρόσφατες εξελίξεις στα μοντέλα που βασίζονται σε μετασχηματιστές προσφέρουν σημαντικές εναλλακτικές λύσεις στα παραδοσιακά συνελικτικά νευρωνικά δίκτυα (CNN), ιδιαίτερα στην καταγραφή εξαρτήσεων και πληροφοριών μεγάλης εμβέλειας. Με κίνητρο τη διαθεσιμότητα αεροφωτογραφιών υψηλής ανάλυσης της ελληνικής ακτογραμμής από το Ελληνικό Κτηματολόγιο, η έρευνα αυτή εστιάζει στην εφαρμογή μοντέλων μετασχηματιστών τελευταίας τεχνολογίας για την επίτευξη σημασιολογικής κατάτμησης στις εικόνες. Αξιοποιώντας προϋπάρχοντα επισημασμένα σύνολα δεδομένων από την ακτογραμμή των ΗΠΑ, προσαρμόζουμε και εκπαιδεύουμε τα μοντέλα SegFormer, MaskFormer και Mask2Former για να οριοθετήσουμε διάφορες κατηγορίες επιφανειών γης, όπως υδάτινα σώματα, βλάστηση, παραλίες και ανεπτυγμένες περιοχές. Μέσω της αξιολόγησης των μοντέλων, το Mask2Former αναδεικνύεται ως το μοντέλο με τις κορυφαίες επιδόσεις, επιτυγχάνοντας 85,43% mIoU στο σύνολο δεδομένων της Ελληνικής ακτογραμμής. Η αξιοποίηση της μεταφοράς μάθησης αποδεικνύεται κομβική, καθώς η χρήση προεκπαιδευμένων μοντέλων βελτιώνει σημαντικά τα τελικά μας αποτελέσματα. Αυτή η εργασία αποτελεί ένα σημαντικό βήμα προς την αξιοποίηση τεχνικών υπολογιστικής όρασης για τηλεπισκόπηση σε παράκτια περιβάλλοντα, ανοίγοντας το δρόμο για μελλοντική έρευνα. Οι μελλοντικές κατευθύνσεις έρευνας περιλαμβάνουν την επέκταση των κατηγοριών των κλάσεων, την ανάλυση σε επίπεδο υλικού και την αξιοποίηση της πληροφορίας της τρίτης διάστασης των εικόνων.

Λέξεις Κλειδιά

Όραση Υπολογιστών, Σημασιολογική Κατάτμηση, Μηχανική Μάθηση, Νευρωνικά Δίκτυα, Μετασχηματιστές, SegFormer, MaskFormer, Mask2Former, Παράκτια περιβάλλοντα

Abstract

Coastal regions play a vital role in various socio-economic activities, environmental sustainability, and biodiversity conservation. Efficient tools and methodologies are required for understanding and monitoring these dynamic and ecologically diverse ecosystems. Remote sensing, especially through aerial and satellite imagery, has emerged as a crucial technology for comprehensive coastline analysis. Furthermore, recent advancements in Transformer-based models offer promising alternatives to traditional convolutional neural networks (CNNs), particularly in capturing long-range dependencies and information. Motivated by the availability of high-resolution aerial images of the Greek coastline from the Hellenic Land Registry, this research focuses on applying state-of-the-art transformer models for semantic segmentation. Leveraging pre-existing labeled datasets from the US coastline, we adapt and train SegFormer, MaskFormer, and Mask2Former models to delineate land surface classes such as water bodies, vegetation, sediments and developed areas. Through the evaluation, Mask2Former emerges as the top-performing model, achieving 85.43% mIoU on the Greek Coastline dataset. Transfer learning proves to be vital, highlighting the value of adapting models to specific datasets. This work represents a crucial step towards leveraging computer vision techniques for remote sensing in coastal environments, paving the way for future research. Future research directions include expanding class categories, utilization of altitude information, material-level analysis, and optimization strategies for model training.

Keywords

Computer Vision, Semantic Segmentation, Machine Learning, Neural Networks, Transformers, Transfer Learning, SegFormer, MaskFormer, Mask2Former, Coastal environments

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ.Κόλλια για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης. Επίσης ευχαριστώ ιδιαίτερα την κ.Τζούβελη για την καθοδήγησή της και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου για την στήριξή τους και τις όμορφες στιγμές.

Αθήνα, Απρίλιος 2024

Πελαγία Δρακοπούλου

Contents

List of Figures

List of Tables

Part I

Εκτεταμένη Περίληψη στα Ελληνικά

Chapter 1

Εισαγωγή

1.1 Κίνητρο

Οι παράκτιες περιοχές αντιπροσωπεύουν δυναμικές και οικολογικά πολυμορφικές περιοχές που είναι ζωτικής σημασίας για διάφορες κοινωνικοοικονομικές δραστηριότητες, την περιβαλλοντική βιωσιμότητα και τη διατήρηση της βιοποικιλότητας. Η κατανόηση και η παρακολούθηση αυτών των πολύπλοκων οικοσυστημάτων απαιτεί αποτελεσματικά εργαλεία και μεθοδολογίες ικανές να εξάγουν λεπτομερείς πληροφορίες από μεγάλα και ποικιλόμορφα σύνολα δεδομένων. Η τηλεπισκόπηση, ιδιαίτερα μέσω της χρήσης εναέριων και δορυφορικών εικόνων, έχει αναδειχθεί ως βασική τεχνολογία για το συγκεκριμένο πρόβλημα, προσφέροντας σημαντικές δυνατότητες για ολοκληρωμένη ανάλυση των ακτογραμμών. Η σημασιολογική κατάτμηση είναι μια θεμελιώδης εργασία στην όραση υπολογιστών μέσω της οποίας εξάγονται ουσιαστικές πληροφορίες από τις εικόνες, διαχωρίζοντάς τις σε σημασιολογικά σημαντικές περιοχές [11, 12, 13, 14]. Παράλληλα, οι πρόσφατες εξελίξεις στα μοντέλα που βασίζονται σε Μετασχηματιστές (Transformers) έχουν φέρει επανάσταση στο πεδίο της όρασης υπολογιστών, προσφέροντας ισχυρές εναλλακτικές λύσεις στα παραδοσιακά συνελικτικά νευρωνικά δίκτυα (CNN). Τα μοντέλα μετασχηματιστών όρασης έχουν επιδείξει αξιοσημείωτη απόδοση στην καταγραφή εξαρτήσεων μεγάλης εμβέλειας και πληροφοριών, καθιστώντας τα κατάλληλα για την ανάλυση των παράκτιων εικόνων. Κίνητρο της εργασίας αποτέλεσε η παραχώρηση ενός συνόλου αεροφωτογραφιών υψηλής ανάλυσης της Ελληνικής ακτογραμμής από το Ελληνικό Κτηματολόγιο. Σε συνεργασία με το τμήμα Γεωλογίας και Γεωπεριβάλλοντος του ΕΚΠΑ μελετήσαμε το συγκεκριμένο σύνολο δεδομένων και καταλήξαμε ότι η κατάλληλη ανάλυση και επεξεργασία του μπορεί να προσφέρει πληροφορίες ιδιαίτερα σημαντικές για την γεωμορφολογική καταγραφή του παράκτιου περιβάλλοντος της Ελλάδας και να ενισχύσει την απομακρυσμένη έρευνα σε μία προσπάθεια περιορισμού των δυσκολιών των in-situ αποστολών.

1.2 Σκοπός

Η παρούσα διατριβή αποτελεί το πρώτο αλλά ιδιαίτερα κομβικό βήμα της έρευνας αυτής, το οποίο είναι η εφαρμογή state-of-the-art μοντέλων μετασχηματιστών για τη σημασιολογική κατάτμηση παράκτιων εικόνων. Πιο συγκεκριμένα, τον εντοπισμό διακριτών κατηγοριών επιφάνειας εδάφους, όπως υδάτινα σώματα, βλάστηση, παραλίες και άλλα φυσικά και μη

χαρακτηριστικά σε επίπεδο εικονοστοιχείου. Το βασικό πρόβλημα ήταν ότι οι εικόνες αυτές δεν περιέχουν κάποιου είδους επισήμανση οπότε δεν μπορούν να χρησιμοποιηθούν για την απευθείας εκπαίδευση των μοντέλων. Για τον λόγο αυτό η έρευνά μας επικεντρώνεται στην αξιοποίηση προϋπαρχόντων επισημασμένων συνόλων δεδομένων από την ακτογραμμή των Η.Π.Α. για την εκπαίδευση των μοντέλων μετασχηματιστών και την περαιτέρω προσαρμογή τους στα ειδικά χαρακτηριστικά της ελληνικής ακτογραμμής. Στόχος μας είναι να αξιολογήσουμε την αποτελεσματικότητα των μοντέλων SegFormer, MaskFormer και Mask2Former στην ακριβή οριοθέτηση διαφόρων κατηγοριών επιφάνειας γης.

1.3 Δομή

Στο πρώτο μέρος [2] καλύπτεται το Θεωρητικό Υπόβαθρο στο οποίο εμβαθύνουμε στις θεωρητικές βάσεις της τηλεπισκόπησης, της κατάτμησης εικόνων, της μεταφοράς μάθησης και των μοντέλων που βασίζονται σε μετασχηματιστές. Στο δεύτερο μέρος [3] καλύπτεται το Πειραματικό στάδιο στο οποίο περιγράφουμε αναλυτικά τη μεθοδολογία που χρησιμοποιείται για την απόκτηση και προεπεξεργασία των δεδομένων. Στη συνέχεια, παρουσιάζεται η εκπαίδευση των μοντέλων και η αξιολόγησή τους μέσω των πειραματικών αποτελεσμάτων. Ενώ, τέλος, στον επίλογο [4] ακολουθεί η εξαγωγή των τελικών συμπερασμάτων και των πιθανών μελλοντικών κατευθύνσεων.

Chapter 2

Θεωρητικό Υπόβαθρο

2.1 Τηλεπισκόπηση

Η τηλεπισκόπηση είναι η μέτρηση των ιδιοτήτων των αντικειμένων που βρίσκονται στην επιφάνεια της Γης, χρησιμοποιώντας δεδομένα τα οποία συγκεντρώνονται τόσο από δορυφόρους όσο και από εναέρια μέσα όπως αεροπλάνα, drones κλπ. Αυτή η μη-επεμβατική τεχνική περιλαμβάνει την συγκέντρωση και ανάλυση πληροφορίας από απόσταση, μέσω της αντίδρασης της ηλεκτρομαγνητικής ακτινοβολίας με την επιφάνεια της Γης. Διαφορετικά αντικείμενα και υλικά αντανακλούν, εκπέμπουν και απορροφούν ενέργεια σε διαφορετικά μήκη κύματος, επιτρέποντας στους αισθητήρες να αντλήσουν πληροφορία σχετικά με την σύσταση και τα χαρακτηριστικά τους. Στην παρούσα εργασία, εστιάζουμε στην τηλεπισκόπηση του παράκτιου περιβάλλοντος μέσω δορυφορικών εικόνων και αεροφωτογραφιών 2 διαστάσεων. Η ανάλυση και η εξαγωγή χαρακτηριστικών των εικόνων αυτών και των αντικειμένων που περιλαμβάνουν, θα πραγματοποιηθεί με την χρήση τεχνικών Μηχανικής Μάθησης και πιο συγκεκριμένα Νευρωνικών Δικτύων. Ιδιαίτερα σημαντικός παράγοντας για την ποιότητα της τηλεπισκόπησης αποτελεί η ανάλυση των εικόνων, η οποία με την πάροδο των ετών είναι και υψηλότερη καθώς εξελίσσεται η τεχνολογία των οπτικών αισθητήρων και παράλληλα αυξάνονται οι δορυφορικές αποστολές. Η επιτυχής ανάλυση των δορυφορικών και εναέριων εικόνων μπορεί να αποτελέσει μια ιδιαίτερα χρήσιμη εναλλακτική στις in-situ αποστολές οι οποίες παρουσιάζουν αρκετές δυσκολίες.

2.2 Κατάτμηση Εικόνας

Η κατάτμηση εικόνας αποτελεί μία τεχνική της όρασης υπολογιστών η οποία περιλαμβάνει την τμηματοποίηση μιας εικόνας σε ένα σύνολο μη επικαλυπτόμενων περιοχών των οποίων η ένωση αποτελεί την πλήρη εικόνα [15]. Αυτή η ταξινόμηση των περιοχών, οδηγεί σε μια προηγμένη ανάλυση και κατανόηση του οπτικού περιεχομένου. Στην κατάτμηση εικόνας έχουμε δύο κύρια στοιχεία: τα αντικείμενα (things) και το υλικό (stuff). Τα αντικείμενα αντιστοιχούν σε μετρήσιμα αντικείμενα σε μια εικόνα (π.χ., άνθρωποι, λουλούδια, ζώα κ.λπ.) ενώ το υλικό αντιπροσωπεύει άμορφες περιοχές (ή επαναλαμβανόμενα μοτίβα) παρόμοιου υλικού, το οποίο είναι μη-μετρήσιμο (π.χ., δρόμος, ουρανός και γρασίδι). Η τμηματοποίηση εικόνας έχει αντιμετωπιστεί με διάφορες τεχνικές επεξεργασίας εικόνας κατά τα χρόνια. Μερικές από αυτές τις αρχικές τεχνικές περιλαμβάνουν τη συσταδοποίηση, που

χρησιμοποιείται με ανίχνευση ακμών και περιγράμματος [16], και προσεγγίσεις ιστογράμματος όπως η εξαγωγή χαρακτηριστικών HOG [17] και SIFT [18]. Ωστόσο, η εισαγωγή των Συνελικτικών Νευρωνικών Δικτύων (CNNs) συνέβαλε σημαντικά στην εξέλιξη της τμηματοποίησης εικόνας η οποία μέσω επιβλεπόμενης μάθησης πετυχαίνει πολύ καλύτερα αποτελέσματα. Από τότε έχουν αυξηθεί σημαντικά τα σύνολα δεδομένων για την κατάτμηση, όπως το Cityscapes [19], το ImageNet [20] και το COCO [21]. Συνολικά, η κατάτμηση εικόνας είναι μια πολύπλοκη εργασία όρασης υπολογιστών και πιο απαιτητική από την ταξινόμηση εικόνας καθώς απαιτεί την ταξινόμηση σε επίπεδο pixel και ανίχνευση σχέσεων σε διάφορες κλίμακες. Σε αυτή την κατεύθυνση, τα Νευρωνικά Δίκτυα είναι ικανά να μάθουν μοτίβα διαθέτοντας πληθώρα παραμέτρων που μπορούν να μιμηθούν πολύπλοκες λειτουργίες για την επίτευξη αυτού του σκοπού. Τα μοντέλα μετασχηματιστών, τα οποία εξετάζουμε σε αυτή την εργασία, είναι οι πιο πρόσφατες αρχιτεκτονικές που ασχολούνται με την κατάτμηση, ξεπερνώντας την απόδοση των CNNs. Γενικότερα έχει αποκτήσει σημαντικό ρόλο, ανά τα χρόνια, σε διάφορες εφαρμογές, όπως η απομακρυσμένη ανίχνευση, η κατανόηση τοπίων και τα αυτόνομα συστήματα. Διαφορετικές σημασιολογίες για την ομαδοποίηση pixel, όπως η κατηγορία ή η μελέτη περιπτώσεων, έχουν οδηγήσει σε διαφορετικούς τύπους εργασιών κατάτμησης, όπως η πανοπτική (panoptic), η μελέτη περιπτώσεων (instance) ή η σημασιολογική (semantic) κατάτμηση.

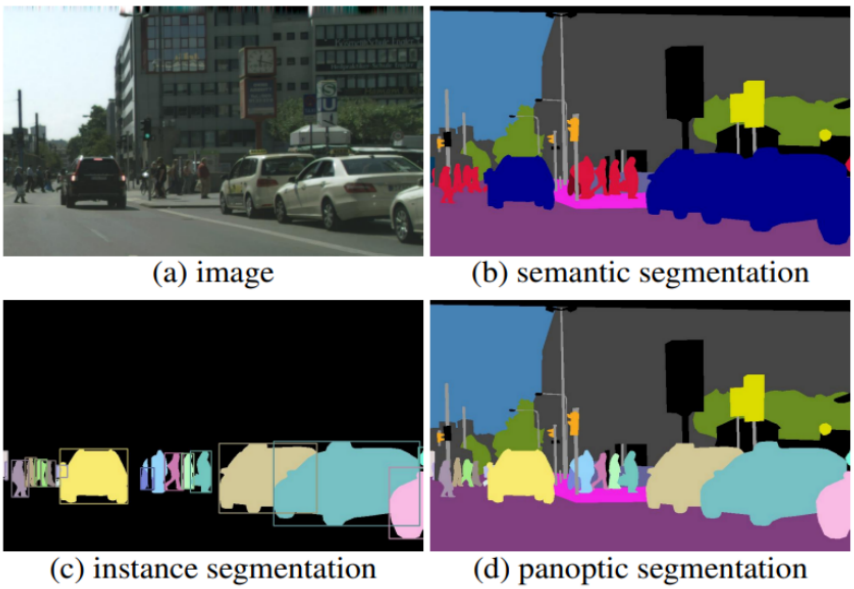


Figure 2.1: *Κατηγορίες κατάτμησης εικόνας*

2.2.1 Σημασιολογική Κατάτμηση Εικόνας

Η σημασιολογική κατάτμηση, η οποία είναι το επίκεντρο αυτής της εργασίας, είναι ένας συγκεκριμένος τύπος τμηματοποίησης εικόνας όπου ο στόχος είναι να αντιστοιχεί μια σημασιολογική ετικέτα σε κάθε εικονοστοιχείο της εικόνας. Πιο συγκεκριμένα, τμηματοποιεί την εικόνα εισόδου σύμφωνα με σημασιολογικές πληροφορίες και προβλέπει τη σημασιολογική κατηγορία κάθε pixel από ένα δεδομένο σύνολο ετικετών. Η έξοδος της σημασιολογικής τμηματοποίησης είναι ένας χάρτης τμηματοποίησης βάσει εικονοστοιχείων, όπου

σε κάθε εικονοστοιχείο εκχωρείται μια ετικέτα κλάσης που αντιπροσωπεύει τον τύπο του αντικειμένου ή της περιοχής στην οποία ανήκει. Η σημασιολογική τμηματοποίηση μελετά τα μη-μετρήσιμα στοιχεία σε μια εικόνα. Αναλύει κάθε εικονοστοιχείο εικόνας και εκχωρεί μια μοναδική ετικέτα κλάσης με βάση την υφή που αντιπροσωπεύει. Η σημασιολογική τμηματοποίηση σχεδιάστηκε για να αναγνωρίζει αντικείμενα που είναι άμορφες περιοχές παρόμοιας υφής ή υλικού [22]. Δεν κάνει διάκριση μεταξύ αντικειμένων της ίδιας κλάσης, απλώς τα ομαδοποιεί. Αυτή η προσέγγιση παρέχει μια υψηλού επιπέδου κατανόηση του περιεχομένου της εικόνας κατηγοριοποιώντας τα pixel σε προκαθορισμένες κατηγορίες. Ξεκινώντας από τα Πλήρως Συνελικτικά Δίκτυα (FCN), οι περισσότερες προσεγγίσεις σημασιολογικής τμηματοποίησης που βασίζονται σε βαθιά μάθηση διατυπώνουν τη σημασιολογική τμηματοποίηση ως ταξινόμηση ανά εικονοστοιχείο, εφαρμόζοντας μια απώλεια ταξινόμησης (loss) σε κάθε εικονοστοιχείο εξόδου. Η σημασιολογική κατάτμηση μπορεί να θεωρηθεί ως επέκταση της ταξινόμησης εικόνας από το επίπεδο της εικόνας στο επίπεδο των pixel. Πιο πρόσφατες τεχνικές έχουν αποδείξει την αποτελεσματικότητα των αρχιτεκτονικών που βασίζονται σε μετασχηματιστές για τη σημασιολογική τμηματοποίηση.

2.2.2 Κατάτμηση Περιπτώσεων

Η κατάτμηση περιπτώσεων (instance segmentation) είναι μια σύνθετη εργασία υπολογιστικής όρασης που αποσκοπεί στην ταυτόχρονη ταξινόμηση και τμηματοποίηση κάθε μεμονωμένης περίπτωσης αντικειμένου σε μια εικόνα. Σε αντίθεση με τη σημασιολογική τμηματοποίηση, η οποία αποδίδει μια κοινή ετικέτα σε όλα τα εικονοστοιχεία που ανήκουν σε μια συγκεκριμένη κατηγορία αντικειμένων, η τμηματοποίηση περιπτώσεων παράγει μοναδικές ετικέτες για κάθε περίπτωση αντικειμένου που υπάρχει στην εικόνα. Η έξοδος της κατάτμησης αντικειμένων είναι μια δυαδική μάσκα που περιγράφει με ακρίβεια τα όρια κάθε αντικειμένου στην εικόνα. Το έργο της τμηματοποίησης αντικειμένων είναι αρκετά δύσκολο, καθώς απαιτεί από το μοντέλο να διακρίνει με ακρίβεια μεταξύ διαφορετικών περιπτώσεων αντικειμένων που ανήκουν στην ίδια κλάση. Αυτό σημαίνει ότι το μοντέλο πρέπει να είναι σε θέση να τμηματοποιεί μεμονωμένα αντικείμενα που μπορεί να έχουν διαφορετικά μεγέθη, σχήματα, προσανατολισμούς και στάσεις, καθώς και αντικείμενα που μπορεί να είναι μερικώς καλυμμένα ή να επικαλύπτονται από άλλα αντικείμενα.

2.2.3 Πανοπτική Κατάτμηση

Η πανοπτική κατάτμηση αφορά την ταυτόχρονη εκτέλεση σημασιολογικής και περιπτωσιολογικής κατάτμησης σε μια εικόνα. Ο στόχος της πανοπτικής κατάτμησης είναι η δημιουργία μιας ενιαίας μάσκας τμηματοποίησης σε επίπεδο εικόνας που συνδυάζει τόσο σημασιολογική όσο και περιπτωσιολογική πληροφορία. Τα μοντέλα πανοπτικής τμηματοποίησης χρησιμοποιούν συνήθως μια διαδικασία δύο σταδίων που αποτελείται από την ανίχνευση αντικειμένων και την τμηματοποίηση για να πετύχουν αυτό το έργο. Το πρώτο στάδιο περιλαμβάνει τη χρήση ενός δικτύου ανίχνευσης αντικειμένων για τον εντοπισμό και την ταξινόμηση διαφορετικών περιπτώσεων αντικειμένων στην εικόνα, ενώ το δεύτερο στάδιο περιλαμβάνει τη χρήση ενός δικτύου κατάτμησης για τη δημιουργία μασκών τμηματοποίησης σε επίπεδο περιπτώσεων για κάθε εντοπισμένη περίπτωση αντικειμένου.

2.3 Μεταφορά Μάθησης

Η θεμελιώδης ιδέα πίσω από τη μεταφορά μάθησης (transfer learning) είναι η εφαρμογή της γνώσης που αποκτάται από εργασίες με επαρκή επισημασμένα δεδομένα (labeled data) σε καταστάσεις όπου είναι διαθέσιμα μόνο περιορισμένα από αυτά. Η δημιουργία επισημασμένων δεδομένων μπορεί να είναι ιδιαίτερα δαπανηρή σε χρόνο, επομένως η αποτελεσματική χρήση των υπαρχόντων συνόλων δεδομένων καθίσταται πολύ σημαντική. Η μεταφορά μάθησης εφαρμόζεται σε μεγάλο βαθμό στους Μετασχηματιστές καθώς τα μοντέλα αυτά χρειάζονται μεγάλα σύνολα δεδομένων για εκπαίδευση, λόγω του βάθους και της πολυπλοκότητας της αρχιτεκτονικής τους. Στα παραδοσιακά μοντέλα μηχανικής μάθησης, ο πρωταρχικός στόχος είναι η γενίκευση σε μη ορατά δεδομένα με βάση τα μοτίβα που μαθαίνονται κατά τη διάρκεια της εκπαίδευσης. Στη μεταφορά μάθησης, ο στόχος είναι να ξεκινήσει αυτή η διαδικασία γενίκευσης ξεκινώντας με μοτίβα που μαθαίνονται για μια διαφορετική εργασία. Αντί να ξεκινά η διαδικασία μάθησης από το μηδέν, όπως συμβαίνει συχνά με τα τυχαία αρχικοποιημένα μοντέλα, η μεταφορά μάθησης ξεκινά με μοτίβα που αποκτήθηκαν προηγουμένως για την αντιμετώπιση μιας διαφορετικής εργασίας. Η τεχνική αυτή είναι απαραίτητη στη σφαίρα της μάθησης, προσομοιώνοντας το χαρακτηριστικό των ανθρώπων οι οποίοι δεν χρειάζεται να διδάσκονται ρητά από το μηδέν κάθε εργασία για να πετύχουν, αλλά αξιοποιούν γνώσεις και δεξιότητες που έχουν αποκομίσει από άλλες παραπλήσιες εργασίες. Τα άτομα αντιμετωπίζουν συχνά νέες καταστάσεις, αλλά προσαρμόζονται και λύνουν προβλήματα συνδυαστικά. Ως εκ τούτου, η μεταφορά μάθησης αντικατοπτρίζει την ικανότητα συλλογής γνώσεων από διάφορες εμπειρίες και την εφαρμογή αυτής της «γνώσης» σε νέα περιβάλλοντα. Η μεταφορά της γνώσης είναι εφικτή μόνο όταν είναι «κατάλληλη», αλλά ο ακριβής καθορισμός της καταλληλότητας σε αυτό το πλαίσιο είναι πρόκληση και συχνά απαιτεί πειραματισμό. Στην παρούσα εργασία, θα βασιστούμε σε μεγάλο βαθμό στα πλεονεκτήματα της τεχνική αυτής, χρησιμοποιώντας προεκπαιδευμένα μοντέλα (transformers) σε μεγάλα datasets, όπως το Cityscapes [19] και το ADE [23] και στη συνέχεια θα τα προσαρμόσουμε στα παραλιακά δεδομένα κάνοντας fine tuning. Το fine tuning είναι μία τεχνική της μεταφοράς μάθησης κατά την οποία η διαδικασία της εκπαίδευσης ξεκινά έχοντας αρχικοποιήσει τα βάρη με βάση αυτά του προεκπαιδευμένου μοντέλου και αντικαθιστώντας πλήρως την κεφαλή ταξινόμησης προσαρμόζοντάς την στις κλάσεις της νέας εργασίας. Στη συνέχεια, κατά την εκπαίδευση αλλάζει σε κάθε εποχή τα βάρη των κρυμμένων επιπέδων επιτυγχάνοντας καλύτερη απόδοση στα νέα δεδομένα. Το γεγονός ότι η εκπαίδευση του μοντέλου δεν ξεκινά από το μηδέν αλλά τα βάρη του είναι αρχικοποιημένα, και μάλιστα για ένα task παρόμοιο με το δικό μας, βελτιώνει κατά πολύ την απόδοση του τελικού μοντέλου και συνεισφέρει στην εξοικονόμηση χρόνου και πόρων.

2.4 Μετασχηματιστής

2.4.1 Αρχιτεκτονική

Ο Μετασχηματιστής (Transformer) είναι μια απλή αρχιτεκτονική δικτύου που βασίζεται εξ ολοκλήρου σε έναν μηχανισμό προσοχής (attention mechanism) για να αντλεί καθολικές

εξαρτήσεις μεταξύ εισόδου και εξόδου [1]. Οι μετασχηματιστές έχουν σχεδιαστεί για να λύσουν το πρόβλημα των RNNs τα οποία αντιμετωπίζουν περιορισμούς μνήμης σε μεγαλύτερα μήκη ακολουθίας. Οι μηχανισμοί προσοχής επιτρέπουν τη μοντελοποίηση των εξαρτήσεων χωρίς να λαμβάνεται υπόψη η απόστασή τους στις ακολουθίες εισόδου ή εξόδου. Η αυτοπροσοχή (self-attention) είναι ένας μηχανισμός προσοχής που συνδέει διαφορετικές θέσεις μιας μεμονωμένης ακολουθίας προκειμένου να υπολογίσει μια αναπαράσταση της ακολουθίας. Ο μετασχηματιστής ήταν το πρώτο μοντέλο μεταγωγής που βασίζεται εξ ολοκλήρου στην αυτοπροσοχή για τον υπολογισμό των αναπαραστάσεων της εισόδου και της εξόδου του χωρίς τη χρήση RNN ή συνέλιξης. Οι Μετασχηματιστές (Transformers) έχει αποδειχθεί ότι υπερτερεί των αλυσιδωτών και συνελικτικών μοντέλων στην επεξεργασία φυσικής γλώσσας, ενώ παρουσιάζει μικρότερη υπολογιστική πολυπλοκότητα και, συνεπώς, ταχύτερο χρόνο εκπαίδευσης. Ενώ στα CNNs απαιτούνται πολλά στοιβαγμένα επίπεδα για τον εντοπισμό εξαρτήσεων μεγάλης εμβέλειας, στον μηχανισμό αυτο-προσοχής των μετασχηματιστών, αυτές οι εξαρτήσεις ανιχνεύονται αποτελεσματικά για μια θέση εξόδου υπολογίζοντας πληροφορίες περιεχομένου από κάθε θέση εισόδου.

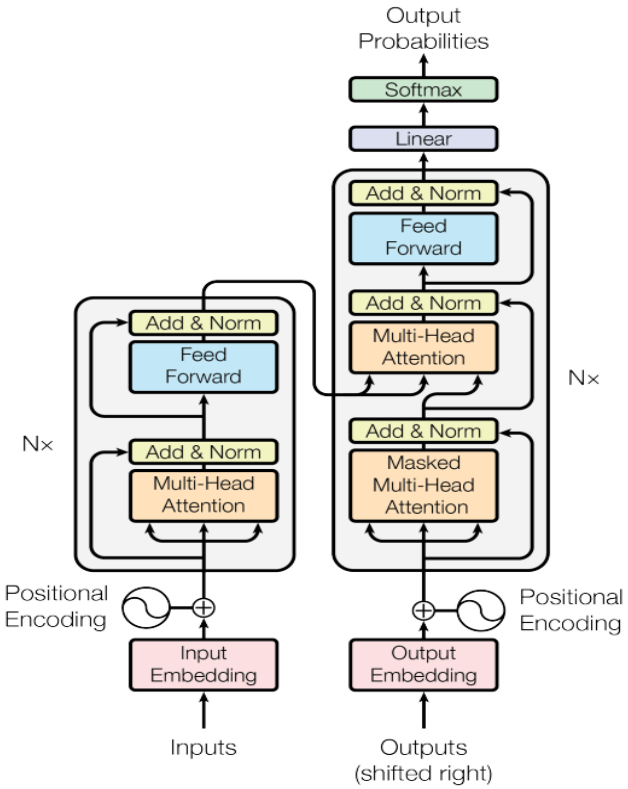


Figure 2.2: *Αρχιτεκτονική Μετασχηματιστή [1]*

Η αρχιτεκτονική ενός μετασχηματιστή βασίζεται σε μια ακολουθία κωδικοποιητών και αποκωδικοποιητών. Ο κωδικοποιητής μετατρέπει αρχικά την ακολουθία εισόδου σε μια διανυσματική αναπαράσταση, γνωστή ως embedding. Συγκεκριμένα, αντιστοιχίζει μια ακολουθία συμβόλων $(x_1, \ldots, x_n)$ σε μια ακολουθία συνεχών αναπαραστάσεων $z = (z_1, \ldots, z_n)$. Αυτή η ακολουθία τροφοδοτείται στη συνέχεια στον αποκωδικοποιητή, ο οποίος δημιουργεί μια

ακολουθία εξόδου από σύμβολα $(y_1, ..., y_n)$, παράγοντας ένα στοιχείο κάθε φορά. Το μοντέλο χρησιμοποιεί σύμβολα που δημιουργήθηκαν προηγουμένως ως πρόσθετη είσοδο κατά τη δημιουργία του επόμενου. Σε κάθε βήμα, ο αποκωδικοποιητής παίρνει ως εισόδους τα embeddings της ακολουθίας εισόδου και τα embeddings της προηγούμενης εξόδου, μετατοπισμένα κατά ένα στοιχείο, προβλέποντας το επόμενο στοιχείο στην ακολουθία.

Κωδικοποιητής: Ο κωδικοποιητής αποτελείται από $N = 6$ στοιβαγμένα επιπέδα, το καθένα από τα οποία περιλαμβάνει δύο υποστρώματα. Το πρώτο είναι ένας μηχανισμός αυτοπροσοχής πολλών κεφαλών (multi-head self-attention mechanism) και το δεύτερο είναι ένα πλήρως συνδεδεμένο δίκτυο προώθησης (feed-forward network). Κάθε υπο-στρώμα ακολουθείται από μια υπολειμματική σύνδεση (residual connection) και κανονικοποίηση. Με άλλα λόγια, η έξοδος κάθε υποστρώματος δίνεται από τον τύπο $LayerNorm(x + Sublayer(x))$, όπου το $Sublayer(x)$ αντιπροσωπεύει τη συνάρτηση που υλοποιείται από το υπο-στρώμα. Για την υποστήριξη αυτών των υπολειμματικών συνδέσεων, όλα τα υποστρώματα του μοντέλου, καθώς και τα στρώματα ενσωμάτωσης, παράγουν εξόδους διάστασης $d_{model} = 512$.

Αποκωδικοποιητής: Ο αποκωδικοποιητής, αποτελείται επίσης από μια στοίβα $N = 6$ πανομοιότυπων επιπέδων και εισάγει ένα τρίτο υπο-στρώμα επιπλέον των δύο υποστρωμάτων που βρίσκονται σε κάθε επίπεδο κωδικοποιητή. Αυτό το πρόσθετο υπο-στρώμα εκτελεί προσοχή πολλαπλών κεφαλών (multi-head attention) στην έξοδο της στοίβας του κωδικοποιητή. Παρόμοια με τον κωδικοποιητή, ο αποκωδικοποιητής ενσωματώνει υπολειμματικές συνδέσεις γύρω από κάθε υπο-στρώμα, ακολουθούμενες από κανονικοποίηση στρώματος. Το υπο-στρώμα αυτοπροσοχής στη στοίβα του αποκωδικοποιητή τροποποιείται για να αποτρέπει την μετάβαση θέσεων σε επόμενες θέσεις. Η αυτο-προσοχή μάσκας περιορίζει την εμβέλεια της αυτο-προσοχής, ώστε ο υπολογισμός των εξόδων να μην επηρεάζεται από τα "μελλοντικά" αποτελέσματα, δηλαδή οι προβλέψεις για τη θέση i εξαρτώνται μόνο από γνωστές εξόδους σε θέσεις μικρότερες από i.

2.4.2 Μηχανισμός Προσοχής

Στα πλαίσια της βαθιάς μάθησης και των νευρωνικών δικτύων ο μηχανισμός Attention αποτελεί μία τεχνική η οποία επιτρέπει στο μοντέλο να εστιάσει επιλεκτικά σε συγκεκριμένα τμήματα των δεδομένων εισόδου, όταν κάνει προβλέψεις ή εξάγει πληροφορίες. Ειδικότερα, παράγει ένα σταθμισμένο άθροισμα το οποίο αντικατοπτρίζει την σημασία των στοιχείων που αποτελούν τα δεδομένα εισόδου.

Scaled Dot Product Attention

Query (Q): Το query αντιπροσωπεύει το στοιχείο στην ακολουθία εισόδου για το οποίο θέλουμε να υπολογίσουμε την σημασία του ή την σχέση εξάρτησής του αναφορικά με τα υπόλοιπα αντικείμενα. Στα πλαίσια του attention mechanism, συνήθως υπάρχει ένα query vector συνδεδεμένο με κάθε θέση στην ακολουθία εισόδου.

Key (K): Το key αντιπροσωπεύει τα στοιχεία στην ακολουθία εισόδου με τα οποία συγκρίνεται το query. Τα key vectors χρησιμοποιούνται για να προσδιορίσουν πόσο καλά κάθε στοιχείο της ακολουθίας εισόδου συσχετίζεται με το query. Όμοια με τα queries, υπάρχει συνήθως ένα key vector για κάθε θέση της ακολουθίας εισόδου.

Value (V): Το value αντιπροσωπεύει την πληροφορία που σχετίζεται με κάθε στοιχείο της ακολουθίας εισόδου. Τα value vectors αποτελούν τα κομμάτια πληροφορίας που σταθμίζονται και συνδυάζονται με βάση τα attention scores που λαμβάνονται από το query και τα key vectors. Όπως με τα queries και τα keys, υπάρχει ένα value vector που σχετίζεται με κάθε θέση στην ακολουθία εισόδου.

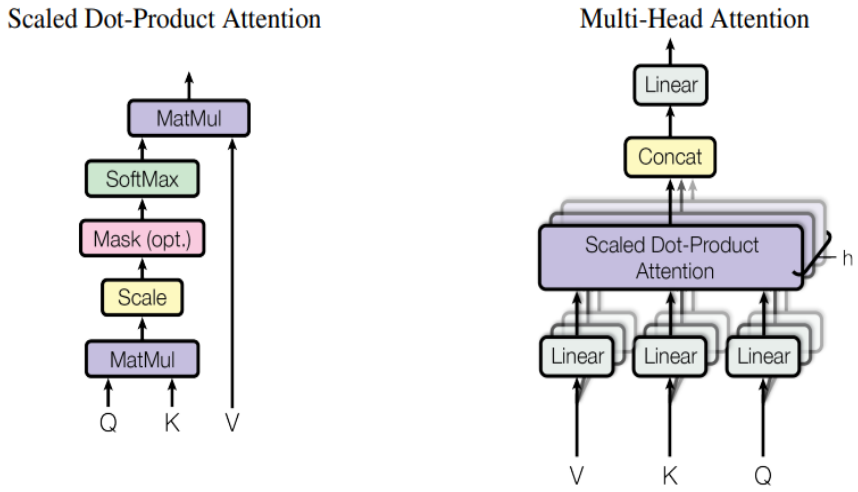


Figure 2.3: *Scaled Dot-Product και Multi-Head Attention Αρχιτεκτονικές [1]*

Τρόπος λειτουργίας του attention mechanism:

1. Για ένα δεδομένο query, ο attention mechanism υπολογίζει ένα σύνολο από attention scores λαμβάνοντας το εσωτερικό γινόμενο του query vector με κάθε key vector στην ακολουθία. Αυτές οι βαθμολογίες αντικατοπτρίζουν την ομοιότητα ή τη συνάφεια κάθε στοιχείου στην ακολουθία με το query.

2. Τα attention scores συνήθως κλιμακώνονται και περνούν μέσω μιας συνάρτησης softmax για να ληφθούν τα κανονικοποιημένα βάρη. Αυτά τα βάρη υποδεικνύουν πόσο πρέπει να εστιάσει το μοντέλο σε κάθε στοιχείο της ακολουθίας σε σχέση με το query.

3. Τα βάρη προσοχής χρησιμοποιούνται για τον γραμμικό συνδυασμό των value vectors, παράγοντας μια έξοδο. Αυτή η έξοδος αντιπροσωπεύει το αποτέλεσμα της εστίασης του μηχανισμού προσοχής στα σχετικά στοιχεία της ακολουθίας εισόδου σε σχέση με το query.

Η βασική ιδέα είναι ότι τα query, key και value vectors επιτρέπουν στον μηχανισμό προσοχής να μαθαίνει δυναμικά και να προσαρμόζεται στο συγκεκριμένο πλαίσιο και τις σχέσεις των δεδομένων, επιτρέποντας στο μοντέλο να καταγράφει περίπλοκες εξαρτήσεις και μοτίβα. Αυτό είναι ένα θεμελιώδες στοιχείο των transformer models που χρησιμοποιούνται τόσο στην επεξεργασία φυσικής γλώσσας όσο και στην όραση υπολογιστών.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2.1}$$

όπου d_k η διαστατικότητα των διανυσμάτων κλειδιών

Multi-Head Attention

Ο μηχανισμός multi-head attention αποτελεί μία επέκταση του scaled dot-product attention μηχανισμού που χρησιμοποιείται στους transformers. Επιτρέπει στο μοντέλο να αναγνωρίζει διαφορετικούς τύπους σχέσεων και εξαρτήσεων μεταξύ των δεδομένων, εφαρμόζοντας τον μηχανισμό attention πολλαπλές φορές παράλληλα με διαφορετικά σύνολα ερωτημάτων, κλειδιών και τιμών. Η βασική διαφορά του με τον scaled dot-product attention μηχανισμό είναι ότι παρέχει μία πιο σύνθετη αναπαράσταση των δεδομένων. Πιο συγκεκριμένα, ο μηχανισμός multi-head attention επιτρέπει στο μοντέλο να παρακολουθεί διαφορετικά μέρη των δεδομένων εισόδου σε διαφορετικές θέσεις, γεγονός που μπορεί να είναι επωφελές για εργασίες όπως η επεξεργασία φυσικής γλώσσας, όπου η σειρά των στοιχείων στα δεδομένα εισόδου είναι σημαντική. Ο multi-head attention χρησιμοποιείται επίσης σε μοντέλα transformer, όπου χρησιμοποιείται για τον υπολογισμό self-attention μεταξύ των στοιχείων της ακολουθίας εισόδου. Δεδομένων των πινάκων Q, K και V , με τους αντίστοιχους πίνακες βαρών W_q, W_k, W_v , μπορούμε να ορίσουμε την προσοχή πολλαπλών κεφαλών με H κεφαλές προσοχής ως εξής :

$$MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_H)W_o \tag{2.2}$$

όπου το κάθε $head_i$ ορίζεται ως:

$$head_i = Attention(QW_{q,i}, KW_{k,i}, VW_{v,i}) \tag{2.3}$$

Concat: συνένωση κατά μήκος της διάστασης των χαρακτηριστικών
W_o: μαθησιακός πίνακας βαρών που εφαρμόζεται στην συνένωση της εξόδου
Q: πίνακας ερωτημάτων
K: πίνακας κλειδιών
V: πίνακας τιμών
W_q, W_k, W_v: πίνακες βαρών που προβάλλουν τα Q, K, V σε έναν κοινό χώρο
Ο μηχανισμός προσοχής πολλαπλών κεφαλών επιτρέπει στο μοντέλο να παρακολουθεί ταυτόχρονα διαφορετικά μέρη της εισόδου, βελτιώνοντας την ικανότητά του να συλλαμβάνει σύνθετα μοτίβα στα δεδομένα.

2.5 Μετασχηματιστές Κατάτμησης

2.5.1 SegFormer

Το SegFormer [2] αποτελεί ένα απλό αλλά αποτελεσματικό framework σημασιολογικής κατάτμησης που ενσωματώνει αποδοτικά τους Μετασχηματιστές με lightweight αποκωδικοποιητές Multilayer Perceptron (MLP). Αυτό το framework εισάγει έναν ιεραρχικά δομημένο κωδικοποιητή Transformer που εξαλείφει την ανάγκη για πρόσθετη κωδικοποίηση θέσης (positional encoding), αποδίδοντας χαρακτηριστικά πολλαπλής κλίμακας. Το μοντέλο SegFormer χρησιμοποιεί έναν αποκωδικοποιητή MLP, ο οποίος συγκεντρώνει στρατηγικά πληροφορίες από διαφορετικά επίπεδα, ενσωματώνοντας τόσο την τοπική όσο και την ευρύτερη

προσοχή (attention). Αυτή η προσέγγιση έχει ως αποτέλεσμα να προκύπτουν ισχυρές αναπαραστάσεις ενώ παράλληλα η απλότητα και ο ελαφρύς σχεδιασμός του SegFormer αποδεικνύονται καθοριστικά για την επίτευξη αποτελεσματικής σημασιολογικής κατάτμησης.

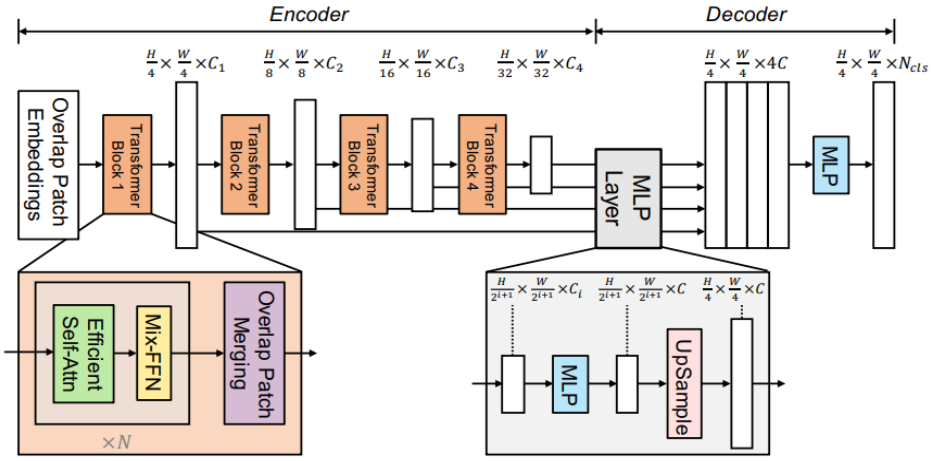


Figure 2.4: *Η αρχιτεκτονική του SegFormer [2]*

Όπως φαίνεται στο παραπάνω σχήμα, το SegFormer περιλαμβάνει δύο κύριες ενότητες: Hierarchical Transformer Κωδικοποιητής: Αυτό το τμήμα εξάγει τα γενικά χαρακτηριστικά υψηλής ανάλυσης και τα πιο λεπτομερή χαρακτηριστικά χαμηλής ανάλυσης. Σε αντίθεση με την προσέγγιση του ViT, η οποία χρησιμοποιεί patches μεγέθους 16×16, το SegFormer διαιρεί μια εικόνα εισόδου μεγέθους $H \times W \times 3$ σε μικρότερα patches μεγέθους 4×4 . Η χρήση αυτών των μικρότερων patches είναι ιδιαίτερα επωφελής για εργασίες πυκνής πρόβλεψης. Ο ιεραρχικός κωδικοποιητής επεξεργάζεται αυτές τις ενημερώσεις κώδικα, παράγοντας λειτουργίες πολλαπλών επιπέδων στα $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ της αρχικής ανάλυσης εικόνας. Light-weight αποκωδικοποιητής All-MLP: Αυτό το τμήμα συγχωνεύει τα χαρακτηριστικά πολλαπλών επιπέδων που λαμβάνονται από τον κωδικοποιητή για να δημιουργήσει την τελική μάσκα σημασιολογικής τμηματοποίησης. Οι ενημερώσεις κώδικα εικόνας λειτουργούν ως είσοδοι στον ιεραρχικό κωδικοποιητή Transformer, με αποτέλεσμα τις λειτουργίες πολλαπλών επιπέδων που στη συνέχεια μεταβιβάζονται στον αποκωδικοποιητή All-MLP. Ο αποκωδικοποιητής προβλέπει τη μάσκα τμηματοποίησης σε ανάλυση $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$, όπου το N_{cls} αντιπροσωπεύει τον αριθμό των κατηγοριών. Ο συνδυασμός ενός ιεραρχικά δομημένου κωδικοποιητή Transformer και ενός light-weight αποκωδικοποιητή All-MLP διακρίνει τη σχεδίαση του SegFormer, επιδεικνύοντας την αποτελεσματικότητά του στην αντιμετώπιση των προκλήσεων της σημασιολογικής τμηματοποίησης.

2.5.2 MaskFormer

Το MaskFormer εισάγει ένα απλό μοντέλο ταξινόμησης μάσκας σχεδιασμένο να προβλέπει ένα σύνολο δυαδικών μασκών, καθεμία συνδεδεμένη με μια μοναδική πρόβλεψη ετικέτας παγκόσμιας κλάσης [3]. Η θεμελιώδης πρωτοπορία του μοντέλου έγκειται στην ευελιξία της ταξινόμησης μάσκας, προσφέροντας μια ενοποιημένη λύση τόσο για εργασίες τμηματοποίησης σε επίπεδο σημασιολογίας όσο και σε instance επίπεδο μέσω ενός κοινού

μοντέλου με κοινή διαδικασίας εκπαίδευσης. Συγκεκριμένα, το MaskFormer επιδεικνύει ανώτερη απόδοση σε σχέση με τις βασικές γραμμές ταξινόμησης ανά pixel, ιδιαίτερα σε σενάρια με μεγάλο αριθμό κλάσεων. Ενώ πολλές προσεγγίσεις σημασιολογικής τμηματοποίησης που βασίζονται σε βαθιά μάθηση αντιμετωπίζουν το πρόβλημα ως ταξινόμηση ανά εικονοστοιχείο, εφαρμόζοντας μια απώλεια ταξινόμησης σε μεμονωμένα εικονοστοιχεία εξόδου, το MaskFormer υιοθετεί μια εναλλακτική πρακτική. Ξεχωρίζει τις πτυχές κατάτμησης και ταξινόμησης προβλέποντας ένα σύνολο δυαδικών μασκών, καθεμία από τις οποίες σχετίζεται με μια πρόβλεψη κλάσης. Αξιοποιώντας τον μηχανισμό πρόβλεψης συνόλου από το DETR, το MaskFormer χρησιμοποιεί έναν αποκωδικοποιητή Transformer για τον υπολογισμό ζευγών, το καθένα από τα οποία περιλαμβάνει μια πρόβλεψη κλάσης και ένα διάνυσμα ενσωμάτωσης μάσκας (mask embedding). Το διάνυσμα ενσωμάτωσης μάσκας διευκολύνει την πρόβλεψη δυαδικής μάσκας μέσω ενός εσωτερικού γινομένου με την ενσωμάτωση ανά εικονοστοιχείο που λαμβάνεται από ένα πλήρως συνελικτικό δίκτυο (FCN).

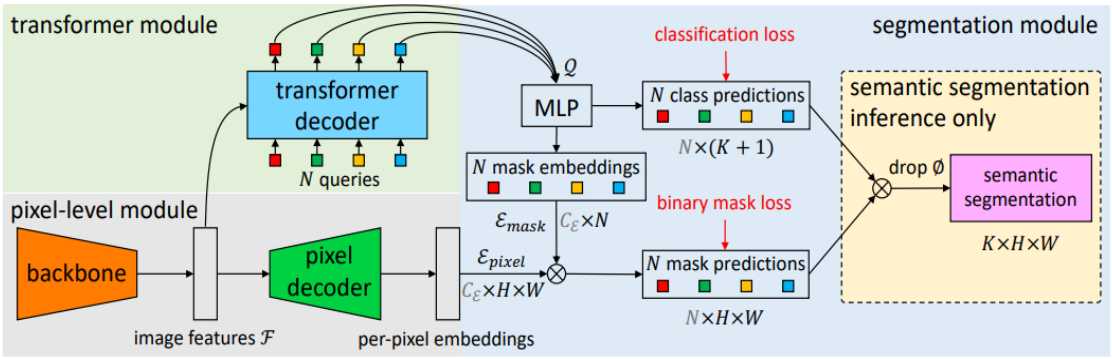


Figure 2.5: *Η αρχιτεκτονική του MaskFormer [3]*

Το MaskFormer υπολογίζει ένα σύνολο από N ζεύγη πιθανοτήτων-μάσκας, που συμβολίζονται ως $z = \{(p_i, m_i)\}_{i=1}^{N}$. Το μοντέλο αποτελείται από τρεις βασικές ενότητες.

Pixel-Level Module: Εξάγει embeddings ανά pixel που είναι πολύ σημαντικά για τη δημιουργία προβλέψεων δυαδικής μάσκας.

Transformer module: Μια στοίβα στρωμάτων αποκωδικοποιητή μετασχηματιστή υπολογίζει N embeddings ανά τμήμα, συμβάλλοντας στα επόμενα βήματα πρόβλεψης.

Segmentation Module: Είναι υπεύθυνο για τη δημιουργία προβλέψεων $\{(p_i, m_i)\}_{i=1}^{N}$ από τα ληφθέντα embeddings. Αυτή η ενότητα εξασφαλίζει μια συνεκτική ενοποίηση προβλέψεων τάξης και δυαδικών μασκών. Η αρχιτεκτονική του MaskFormer επιδεικνύει τον συνδυασμό αυτών των λειτουργικών μονάδων, επιδεικνύοντας την αποτελεσματικότητά της στην αντιμετώπιση εργασιών τμηματοποίησης τόσο σε σημασιολογικό όσο και σε επίπεδο παρουσίασης μέσα σε ένα ενοποιημένο πλαίσιο.

2.5.3 Mask2Former

Το Masked-attention Mask Transformer (Mask2Former) [4] αποτελεί μια ευέλικτη αρχιτεκτονική σχεδιασμένη για την αντιμετώπιση διαφόρων εργασιών τμηματοποίησης εικόνων, συμπεριλαμβανομένης της πανοπτικής, της instance και της σημασιολογικής. Κλειδί για την απόδοσή του είναι οι μηχανισμοί συγκαλυμμένης προσοχής (masked attention) που εξάγουν τοπικά χαρακτηριστικά περιορίζοντας τη, cross attention σε προβλεπόμενες περιοχές μάσκας. Το Mask2Former ξεπερνά τις εξειδικευμένες αρχιτεκτονικές σε διάφορα task κατάτμησης, εξασφαλίζοντας ευκολία εκπαίδευσης. Αποτελούμενο από έναν εξαγωγέα χαρακτηριστικών (feature extractor) κορμού, έναν αποκωδικοποιητή pixel και έναν αποκωδικοποιητή Transformer, το Mask2Former εισάγει αρκετές βελτιώσεις για καλύτερα αποτελέσματα και αποτελεσματική εκπαίδευση.

Η πρώτη βελτίωση περιλαμβάνει τη χρήση του μηχανισμού masked attention στον αποκωδικοποιητή Transformer, περιορίζοντας την προσοχή σε χαρακτηριστικά που επικεντρώνονται γύρω από τα προβλεπόμενα τμήματα. Αυτή η τοπική εστίαση, είτε σε αντικείμενα είτε σε περιοχές, έρχεται σε αντίθεση με τον μηχανισμό cross attention ενός τυπικού αποκωδικοποιητή Transformer, οδηγώντας σε ταχύτερη σύγκλιση και βελτιωμένη απόδοση. Δεύτερον, το Mask2Former αξιοποιεί χαρακτηριστικά υψηλής ανάλυσης πολλαπλής κλίμακας, ενισχύοντας την ικανότητά του να τμηματοποιεί μικρά αντικείμενα ή περιοχές. Τρίτον, εισάγει ορισμένες βελτιστοποιήσεις, όπως η αντιστροφή της σειράς της αυτο-προσοχής (self-attention) και του cross-attention, η δυνατότητα εκμάθησης των χαρακτηριστικών ερωτημάτων (queries) και η κατάργηση του dropout, συμβάλλουν στη βελτιωμένη απόδοση χωρίς πρόσθετο υπολογιστικό κόστος. Τέλος, το μοντέλο επιτυγχάνει μια 3× μείωση στη μνήμη εκπαίδευσης χωρίς να διακυβεύεται η απόδοση υπολογίζοντας την απώλεια μάσκας σε μερικά σημεία τυχαίας δειγματοληψίας. Αυτές οι βελτιώσεις όχι μόνο αυξάνουν την απόδοση του μοντέλου αλλά και απλοποιούν σημαντικά την εκπαίδευση, καθιστώντας τις καθολικές αρχιτεκτονικές πιο προσιτές σε χρήστες με περιορισμένους υπολογιστικούς πόρους.

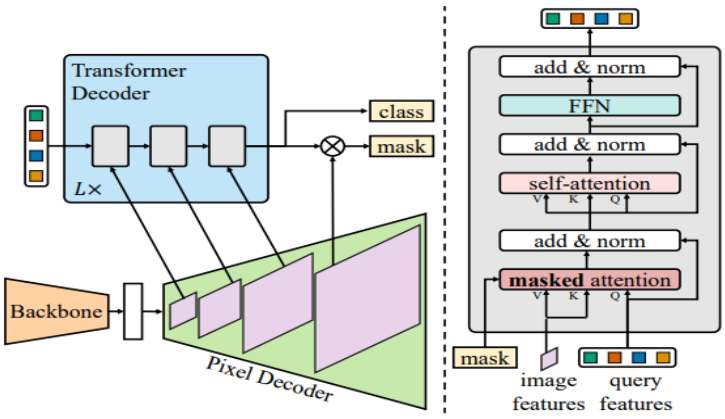


Figure 2.6: *Η αρχιτεκτονική του Mask2Former [4]*

Το Mask2Former υιοθετεί την ίδια μετα-αρχιτεκτονική με το MaskFormer, που διαθέτει έναν κορμό (backbone), έναν αποκωδικοποιητή pixel και έναν αποκωδικοποιητή Transformer. Σημειωτέον, ο αποκωδικοποιητής Transformer εισάγει masked-attention αντί του

συμβατικού cross-attention. Για την αποτελεσματική αντιμετώπιση μικρών αντικειμένων, μια πρακτική προσέγγιση περιλαμβάνει τη χρήση χαρακτηριστικών υψηλής ανάλυσης από έναν αποκωδικοποιητή pixel, τροφοδοτώντας μια κλίμακα του χαρακτηριστικού πολλαπλής κλίμακας σε ένα στρώμα αποκωδικοποιητή Transformer κάθε φορά.

2.6 Μετρικές

Όπως υποδεικνύεται στο [24], η σημασιολογική κατάτμηση είναι μια πολύπλοκη διαδικασία που εξετάζει τις σχέσεις μεταξύ των ταξινομημένων pixel. Η ακρίβεια σε επίπεδο pixel (pACC) χρησιμεύει ως αρχική μέτρηση για την αξιολόγηση της απόδοσης και υπολογίζεται με την ακόλουθη εξίσωση:

$$acc = \frac{\sum_{i=1}^{k} n_{ij}}{\sum_{i=1}^{k} t_i} \tag{2.4}$$

όπου n_{ij} είναι ο αριθμός των pixel που ανήκουν στην κλάση i και επισημάνθηκαν ως κλάση j, k είναι ο συνολικός αριθμός κλάσεων και $t_i = \sum_{j=1}^{k} n_{ij}$ είναι ο συνολικός αριθμός pixel της κλάσης i. Ωστόσο, αυτή η μέτρηση μπορεί να είναι παραπλανητικά υψηλή σε σύνολα δεδομένων όπου εκτεταμένες περιοχές κυριαρχούνται από μία μόνο κλάση. Αυτό το ζήτημα μπορεί να αντιμετωπιστεί με τις ακόλουθες μετρικές:

mACC Η μέση ακρίβεια είναι η μέση τιμή της ακρίβειας σε όλες τις κατηγορίες:

$$mACC = \frac{1}{k} \sum_{i=1}^{k} \frac{n_{ij}}{t_i} \tag{2.5}$$

IoU Ο λόγος της τομής προς την ένωση (IoU) μετρά την επικάλυψη μεταξύ της προβλεπόμενης σημασιολογικής επισήμανσης και της πραγματικής επισήμανσης για κάθε κλάση, παρέχοντας πληροφορίες για το πόσο καλά ευθυγραμμίζονται οι προβλέψεις του μοντέλου με τα πραγματικά όρια αντικειμένων ή περιοχών στην εικόνα. Η μετρική IoU είναι μια αξιολόγηση ανά κλάση σχετικά με την ομοιότητα της εξαγόμενης κατάτμησης και της πραγματικής, διαιρούμενη με την ένωση:

$$IoU = \frac{TruePositive_i}{TruePositive_i + FalsePositive_i + FalseNegative_i} \tag{2.6}$$

όπου τα $TruePositive_i$, $FalsePositive_i$, και $FalseNegative_i$ είναι τα πλήθη των πραγματικά θετικών, ψευδών θετικών και ψευδών αρνητικών εικονοστοιχείων, αντίστοιχα, για την κατηγορία i.

mIoU Παρέχει ένα μέτρο της συνολικής απόδοσης της τμηματοποίησης υπολογίζοντας τον

μέσο όρο των τιμών IoU σε όλες τις κατηγορίες:

$$mIoU = \frac{1}{k} \sum_{i=1}^{k} IoU_i \tag{2.7}$$

όπου k είναι ο συνολικός αριθμός κλάσεων ή κατηγοριών.

F1-score: Το F1-score είναι ένα μέτρο της ακρίβειας ενός μοντέλου, εξισορροπώντας τόσο το precision όσο και το recall. Υπολογίζεται ως ο αρμονικός μέσος του precision και του recall:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.8}$$

όπου το precision είναι ο λόγος των αληθινών θετικών προβλέψεων προς τον συνολικό αριθμό των θετικών προβλέψεων και το recall είναι ο λόγος των αληθινών θετικών προβλέψεων προς τον συνολικό αριθμό των πραγματικών θετικών προβλέψεων.

Chapter 3

Μέθοδος και Αποτελέσματα

3.1 Προετοιμασία Συνόλων Δεδομένων

3.1.1 Coast Train Dataset

Περιγραφή

Το "Coast Train" [25] είναι ένα σύνολο επισημασμένων δεδομένων που αποτελείται από ορθομωσικές και δορυφορικές εικόνες που καταγράφουν διάφορα παράκτια περιβάλλοντα των Η.Π.Α. Κάθε υποσύνολο δεδομένων στο "Coast Train" σχετίζεται ειδικά με έναν μοναδικό τύπο εικόνας και σύνολο κλάσης. Όσον αφορά τη χωρική ανάλυση, έχουμε από 0,05 m έως 1 m για τα ορθομωσαϊκά, ενώ οι δορυφορικές εικόνες είναι είτε στα 10 m είτε στα 15 m. Ειδικότερα, το σύνολο δεδομένων περιλαμβάνει μια εικόνες από πολλαπλές πηγές, συμπεριλαμβανομένων των NAIP (1m), Quadrangle (6m), Sentinel-2 (10m) και Landsat-8 (15m), προσφέροντας συλλογικά μια καθολική εικόνα για την ποικιλομορφία των παράκτιων περιβαλλόντων. Αυτή η ποικιλόμορφη σειρά αναλύσεων pixel εξασφαλίζει μια ολοκληρωμένη αναπαράσταση των παράκτιων χαρακτηριστικών σε διαφορετικές κλίμακες.

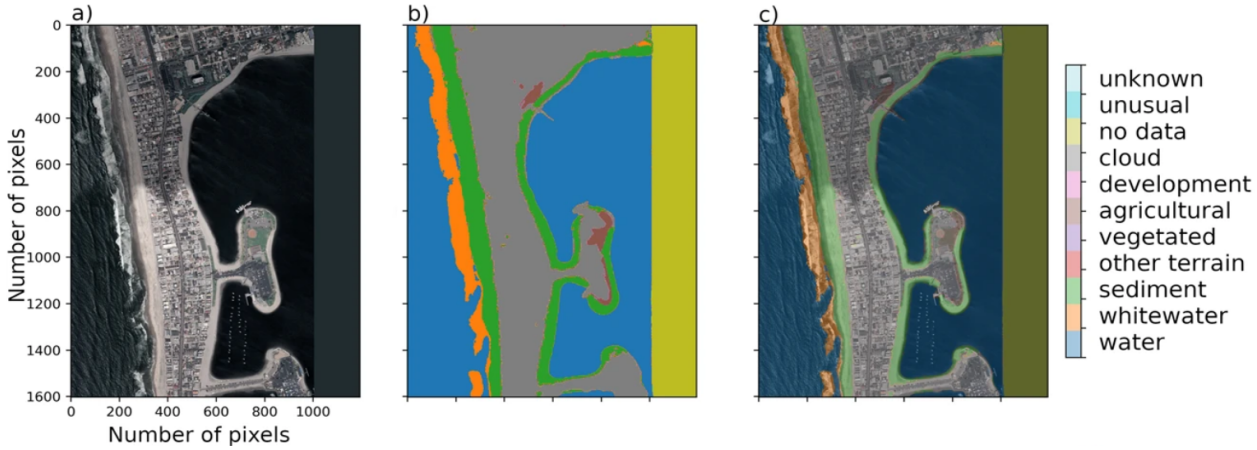


Figure 3.1: *(a):Παράδειγμα φωτογραφίας του Coast Train dataset, (b):Η αντιστοιχη εικόνα επισήμανσης, (c):Φωτογραφία μαζί με τις ετικέτες. Η φωτογραφία ανήκει στο ορθομωσαϊκό σύνολο δεδομένων και προέρχεται από το San Diego, California*

Οι ετικέτες κλάσεων εντός του συνόλου δεδομένων κυμαίνονται μεταξύ 4 και 12, συμβάλλοντας σε έναν λεπτομερή σχολιασμό των εικόνων. Συνολικά, το σύνολο δεδομένων περιλ-

αμβάνει 1852 μεμονωμένες εικόνες, που περιλαμβάνουν 1,196 δισεκατομμύρια pixel. Αυτό το τεράστιο σύνολο δεδομένων αντιπροσωπεύει μια συνολική επιφάνεια 3,63 εκατομμυρίων εκταρίων, παρέχοντας μία πλούσια πηγή για τη μελέτη και την κατανόηση των παράκτιων οικοσυστημάτων. Το γεωγραφικό εύρος του συνόλου δεδομένων είναι εκτεταμένο, και εκτείνεται από 26 έως 48 μοίρες Β σε γεωγραφικό πλάτος και 69 έως 123 μοίρες Δ στο γεωγραφικό μήκος, όπως φαίνεται στο Σχήμα 3.2. Αυτή η ευρεία κάλυψη διασφαλίζει ότι το σύνολο δεδομένων καταγράφει ένα ευρύ φάσμα παράκτιων περιβαλλόντων, καθιστώντας το πολύτιμο πόρο για έρευνα και ανάλυση στο πεδίο.

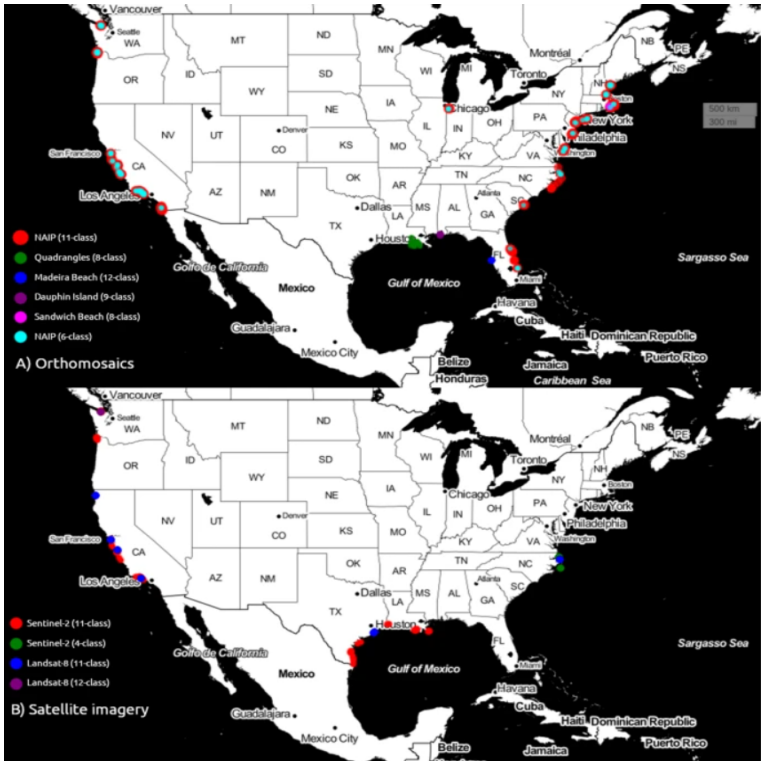


Figure 3.2: *Γεωγραφική κατανομή των (Α) ορθομωσαϊκών και (Β) δορυφορικών εικόνων*

Κάθε εγγραφή δεδομένων έχει ένα μοναδικό σύνολο κλάσεων. Ωστόσο, οι ετικέτες επανεπεξεργάζονται εύκολα για να αντιστοιχίσουν πολλαπλές κλάσεις σε ένα τυποποιημένο σύνολο «υπερκλάσεων» σε όλες τις εγγραφές δεδομένων. Οι υπερκλάσεις, όπως φαίνεται στον Πίνακα 3.1, είναι ονόματα ευρειών κλάσεων για μια συλλογή ετικετών κλάσεων στοιχείων. Για παράδειγμα, τα «κτήρια» και τα «οχήματα» είναι ένα υποσύνολο της «ανεπτυγμένης» υπερκατηγορίας και η «άμμος» και το «χαλίκι» αποτελούν μέρος της υπερκατηγορίας «ιζημάτων». Ορίζονται επτά ετικέτες υπερκλάσεων και μεταξύ τεσσάρων και δώδεκα ετικετών κλάσεων ανάλογα με το σύνολο δεδομένων.

Αντιστοίχιση Υπερκλάσεων	
Ονόματα Υπερκλάσεων	**Κλάσεις**
water	water, sediment plume
whitewater	whitewater, surf
sediment	sediment, sand, gravel, gravel/shell, cobble/boulder/ mud/silt
developed	developed, dev, coastal defense, pavement/road, other anthro, vehicles, buildings, development
natural terrain	bedrock, bare ground, other natural terrain, other bare natural terrain
vegetation	vegetated, vegetated surface, vegetated ground, terrestrial vegetation, marsh vegetation, herbaceous veg, herbaceous vegetation, wood vegetation, woody veg
other	other, unknown, unusual, nodata, people, ice/snow, cloud

Table 3.1: *Αντιστοίχιση των κλάσεων σε υπερκλάσεις.*

Προετοιμασία

Το σύνολο δεδομένων Coast Train περιλαμβάνει δέκα διακριτά υποσύνολα δεδομένων, συγκεκριμένα Landsat8-11-001, Landsat8-12-001, NAIP-11-001, NAIP-6-001, Orthophoto-12-001, Orthophoto-8-001, Orthophoto-9 -001, Quadrangles-7-001, Sentinel2-11-001, και Sentinel2-4-001. Αυτά τα υποσύνολα δεδομένων παρουσιάζουν μια διαφορετική σειρά εικόνων παράκτιου περιβάλλοντος, που ποικίλλουν σε ανάλυση, μέγεθος και περιεχόμενο. Προκειμένου να δημιουργηθεί ένα σύνολο δεδομένων προσαρμοσμένο στις συγκεκριμένες απαιτήσεις της εργασίας σημασιολογικής κατάτμησης, πραγματοποιήθηκε λεπτομερής επιθεώρηση κάθε υποσυνόλου δεδομένων. Σε αυτήν τη διαδικασία, αποκλείσαμε εικόνες με διαστάσεις κάτω των 300x300, καθώς το μικρότερο μέγεθός τους θα μπορούσε να θέσει σε κίνδυνο την ικανότητα του μοντέλου να καταγράφει σημαντικές λεπτομέρειες για την εργασία τμηματοποίησης. Επιπλέον, οι εικόνες που κρίθηκαν μη συναφείς ή δεν συμβάλλουν στους στόχους αυτής της μελέτης αποκλείστηκαν από την επιλογή. Για παράδειγμα, τα υποσύνολα δεδομένων NAIP και Quadrangles εξαιρέθηκαν επειδή θα είχαν αρνητική επίδραση στη διαδικασία εκπαίδευσης, αυξάνοντας τον χρόνο εκπαίδευσης των μοντέλων και μη συνεισφέροντας στα αποτελέσματά τους. Με την παραπάνω διαδικασία, από την αρχική δεξαμενή των 1852 εικόνων, κρατήσαμε 645 εικόνες. Όλες οι επιλεγμένες εικόνες έγιναν resized σε ομοιόμορφη διάσταση 512x512 pixel, για τους σκοπούς της εκπαίδευσης τν μοντέλων. Η παραπάνω διαδικασία προεπεξεργασίας διασφαλίζει ότι το σύνολο δεδομένων coastTrain που χρησιμοποιήθηκε στη μελέτη μας είναι βελτιστοποιημένο για εκπαίδευση μοντέλων μετασχηματιστών σε εργασίες σημασιολογικής κατάτμησης που σχετίζονται με παράκτια περιβάλλοντα, επιτυγχάνοντας μια ισορροπία μεταξύ της συμπερίληψης και της ακρίβειας.

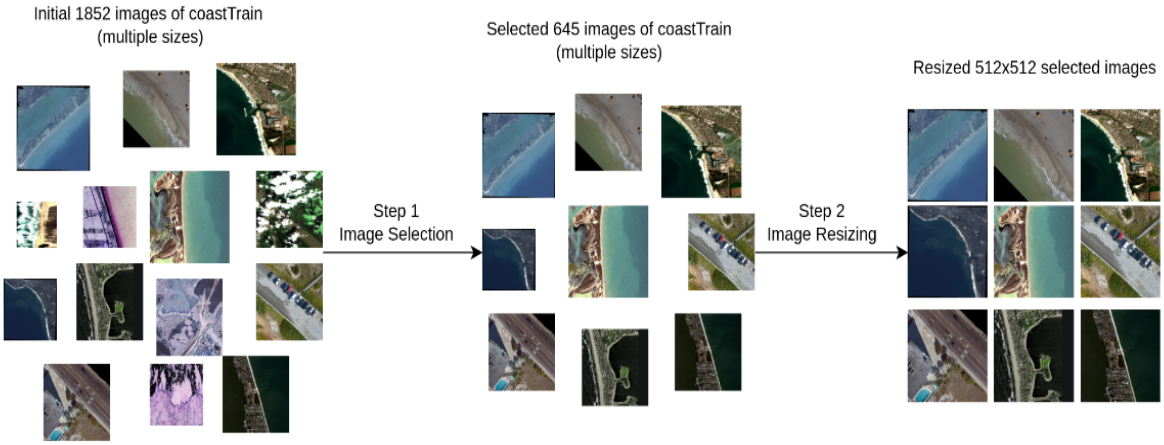


Figure 3.3: *Διαδικασία προεπεξεργασίας του αρχικού Coast Train dataset. Βήμα 1: Επιλέγουμε τις πιο κατάλληλες εικόνες με βάση το μέγεθός, την ανάλυση και την συμβατότητά τους με την παρούσα εργασία. Βήμα 2: Κάνουμε resize όλες τις εικόνες ώστε να έχουν κοινή διάσταση 512x512.*

Το τελικό σύνολο δεδομένων coastTrain αποτελείται από 645 εικόνες, η καθεμία με ανάλυση 512x512 pixel, συνοδευόμενες από τις αντίστοιχες μάσκες τους. Για να διασφαλίσουμε τη συνέπεια στο σύνολο δεδομένων για αποτελεσματική εκπαίδευση, έχουμε αντιστοιχίσει όλες τις κλάσεις στην αντίστοιχη υπερκλάση τους (Πίνακας 3.1). Αυτή η αντιστοίχιση βοηθά στη δημιουργία ενός ενιαίου συνόλου δεδομένων προσαρμοσμένο στις συγκεκριμένες απαιτήσεις της εργασίας. Το σύνολο δεδομένων περιλαμβάνει επτά διακριτές κλάσεις, όπως περιγράφεται στον παρακάτω πίνακα (Πίνακας 3.2). Κάθε pixel στις εικόνες μπορεί να αντιστοιχιστεί σε μία από αυτές τις κατηγορίες, συμβάλλοντας στον πλούτο των πληροφοριών που συλλέγονται για ακριβή σημασιολογική τμηματοποίηση.

Κλάσεις	
Id	**Ετικέτα**
0	water
1	whitewater
2	sediment
3	development
4	natural terrain
5	vegetation
6	unknown

Table 3.2: *Πίνακας αντιστοίχισης id και ετικετών για το τελικό coastTrain dataset.*

3.1.2 Greek Coastline Dataset (Από το Ελληνικό Κτηματολόγιο)

Figure 3.4: *Παράδειγμα εικόνας του Greek Coastline Dataset*

Περιγραφή

Το σύνολο δεδομένων Greek Coastline αποτελείται από εναέριες εικόνες από την Ακτή της Πελοποννήσου σε ανάλυση pixel 25 cm και μας παραχωρήθηκε από το Ελληνικό Κτηματολόγιο. Τα αρχεία δεδομένων είναι σε μορφή raster, επομένως χρησιμοποιήσαμε τη βιβλιοθήκη της python Rasterio [26] για να τα διαβάσουμε και να τα επεξεργαστούμε. Πιο συγκεκριμένα, κάθε εικόνα αποτελείται από τέσσερα ξεχωριστά αρχεία (.tif, .tfw, .aux, .img), ο συνδυασμός των οποίων σχηματίζει την τελική τρισδιάστατη εικόνα (γεωγραφικό μήκος, γεωγραφικό πλάτος, υψόμετρο). Τα αρχεία .tif, .tfw και .aux περιέχουν στοιχεία που συνθέτουν τη δισδιάστατη εικόνα, ενώ το αρχείο .img περιέχει πληροφορίες σχετικά με την τρίτη διάσταση της εικόνας, δηλαδή το ύψος. Συγκεκριμένα, το αρχείο .tif περιλαμβάνει το φάσμα της δισδιάστατης εικόνας σε τέσσερις ξεχωριστές φασματικές ζώνες (Κόκκινο, Πράσινο, Μπλε, Υπέρυθρο). Το αρχείο .tfw περιέχει τα δεδομένα μετασχηματισμού που πρέπει να εφαρμοστούν σε κάθε pixel της εικόνας για να προκύψει το αντίστοιχο γεωγραφικό μήκος και γεωγραφικό πλάτος. Το αρχείο .aux περιλαμβάνει όλα τα μεταδεδομένα που σχετίζονται με τη δισδιάστατη εικόνα, όπως το σύστημα αναφοράς συντεταγμένων που χρησιμοποιείται από την εικόνα για να εκφράσει το γεωγραφικό μήκος και πλάτος των στοιχείων της. Τέλος, το αρχείο .img περιέχει την τρίτη διάσταση της εικόνας, δηλαδή τα δεδομένα για το ύψος κάθε pixel στη δισδιάστατη εικόνα, ενώ περιλαμβάνει και τα αντίστοιχα μεταδεδομένα που σχετίζονται με αυτή τη διάσταση. Το μήκος και το πλάτος της εικόνας ενδέχεται να διαφέρουν ελαφρώς από εικόνα σε εικόνα. Επιλέγουμε να περικόψουμε τα επιπλέον pixel που μπορεί να υπάρχουν σε ορισμένες εικόνες για να έχουμε ένα ενιαίο σύνολο δεδομένων που αποτελείται από εικόνες με διαστάσεις 3200x2400. Η ανάλυση pixel των αρχείων .tif (εικόνες) είναι 25 εκατοστά, ενώ για τα αρχεία .img (ύψος) είναι 20 μέτρα. Έτσι, έχουμε μια αναλογία 1/4 σε κάθε διάσταση μεταξύ αρχείων εικόνας και ύψους. Επομένως, για να επιτύχουμε ευθυγράμμιση στις συντεταγμένες των δύο εικόνων, αναδιαμορφώνουμε τη μήτρα ύψους ώστε να έχει διαστάσεις 3200x2400 αντί για 800x600. Αυτό το επιτυγχάνουμε χρησιμοποιώντας το γινόμενο Kronecker, πολλαπλασιάζοντας κάθε στοιχείο του πίνακα ύψους με έναν πίνακα

γειτονιάς 4x4.

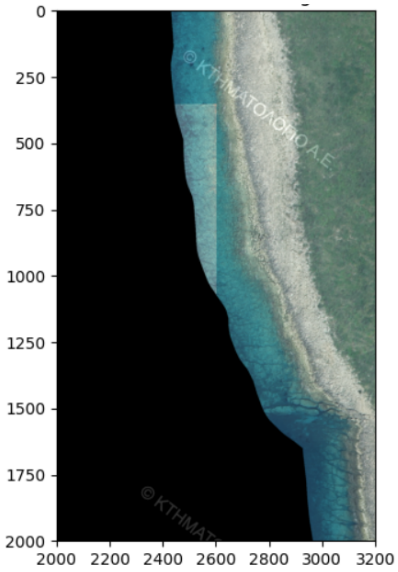


Figure 3.5: *Παράδειγμα 2D εικόνας*

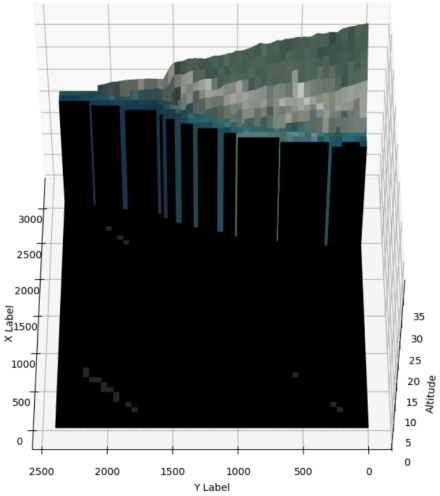


Figure 3.6: *Παράδειγμα 3D εικόνας*

Από το αρχείο .tfw αποκτούμε τον μετασχηματισμό συντεταγμένων που μετατρέπει τις συντεταγμένες των pixel σε πραγματικές γεωγραφικές συντεταγμένες. Πιο συγκεκριμένα, η δομή του μετασχηματισμού φαίνεται στην εξίσωση 3.1 παρακάτω.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} A & B \\ D & -E \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} C \\ F \end{bmatrix} \tag{3.1}$$

Όπου:

- x_1 υπολογίζεται ως η x-συντεταγμένη του εικονοστοιχείου στον χάρτη.
- y_1 υπολογίζεται ως η y-συντεταγμένη του εικονοστοιχείου στον χάρτη.
- x είναι ο αριθμός της στήλης του εικονοστοιχείου στην εικόνα.
- y είναι ο αριθμός της γραμμής του εικονοστοιχείου στην εικόνα.
- A είναι η x-κλίμακα, οι διαστάσεις του εικονοστοιχείου σε μονάδες χάρτη στην x-διεύθυνση.
- B και D είναι οι όροι περιστροφής.
- C και F είναι οι όροι μετάφρασης, αντιπροσωπεύοντας τις x, y συντεταγμένες χάρτη του πάνω αριστερά εικονοστοιχείου.
- E είναι το αρνητικό της y-κλίμακας, οι διαστάσεις του εικονοστοιχείου σε μονάδες χάρτη στην y-διεύθυνση.

Για το σύνολο δεδομένων μας, οι τιμές των σταθερών παραμέτρων A, B, D, E φαίνονται στον πίνακα 3.3 παρακάτω. Οι παράμετροι C και F είναι διαφορετικές για κάθε εικόνα καθώς αντιστοιχούν στο γεωγραφικό μήκος και πλάτος του επάνω αριστερού εικονοστοιχείου.

Παράμετροι Μετασχηματισμού	
Όνομα	**Τιμή**
A	0.25
B	0.00
D	0.00
E	-0.25

Table 3.3: *Παράμετροι μεταχηματισμού για την μετατροπή των συντεταγμένων των εικονοστοιχείων στο σύστημα αναφοράς HGRS87.*

Χρησιμοποιώντας αυτόν τον μετασχηματισμό, οι συντεταγμένες των pixel στην εικόνα αντιστοιχίζονται στο αντίστοιχο γεωγραφικό μήκος και γεωγραφικό πλάτος. Οι γεωγραφικές συντεταγμένες που προκύπτουν εκφράζονται στο Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς 1987 (HGRS87), όπως υποδεικνύεται από τα μεταδεδομένα στο αρχείο .aux.

Προετοιμασία

Για να διασφαλίσουμε την καταλληλότητα των εικόνων για εκπαίδευση σύμφωνα με τους περιορισμούς εισόδου των μοντέλων που θα χρησιμοποιήσουμε, εφαρμόζουμε μια σχολαστική διαδικασία προεπεξεργασίας. Όλα τα μοντέλα που χρησιμοποιήθηκαν στη μελέτη μας χρησιμοποιούν σταθερό μέγεθος εισόδου 512x512 pixel. Ωστόσο, για το σύνολο δεδομένων της ελληνικής ακτογραμμής, υιοθετούμε μια ξεχωριστή προσέγγιση για τη διατήρηση της μέγιστης δυνατής ανάλυσης, δεδομένης της περίπλοκης φύσης του έργου κατάτμησης των ακτών. Για να το πετύχουμε αυτό, επιλέγουμε τον διαχωρισμό εικόνας αντί του resizing, προκειμένου να διατηρήσουμε την υψηλότερη ανάλυση, καθώς είναι ζωτικής σημασίας για τις απαιτήσεις της παράκτιας κατάτμησης. Η συγκεκριμένη διαδικασία προεπεξεργασίας που ακολουθούμε απεικονίζεται οπτικά στο Σχήμα 3.7 και περιλαμβάνει δύο βασικά βήματα. Πρώτον, ξεκινάμε την προεπεξεργασία εφαρμόζοντας zero-padding στις αρχικές εικόνες, οι οποίες έχουν αρχικά μέγεθος 3200x2400 pixel. Το zero-padding εφαρμόζεται τόσο οριζόντια όσο και κάθετα για να διασφαλιστεί ότι οι διαστάσεις διαιρούνται με ακρίβεια με το 512. Έτσι, μετά από αυτό το βήμα το μέγεθος ορίζεται σε 3584x2560 για κάθε εικόνα. Στη συνέχεια, η προκύπτουσα εικόνα χωρίζεται σε 35 εικόνες 512x512.

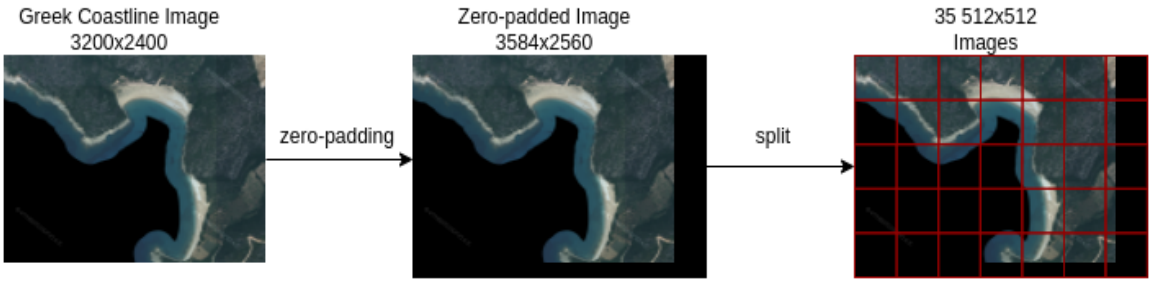


Figure 3.7: *Προεπεξεργασία εικόνας: zero-padding και χωρισμός σε patches*

Figure 3.8: *Παράδειγμα patch*

Ο στόχος αυτής της διπλωματική εργασίας είναι να εξάγει πληροφορίες από το σύνολο δεδομένων της Ελληνικής Ακτογραμμής, το οποίο αρχικά δεν ήταν labeled και περιελάμβανε ακατέργαστες εικόνες. Για να βελτιώσουμε την ποιότητα των αποτελεσμάτων μας, επιλέξαμε να επισημάνουμε χειροκίνητα ένα επιλεγμένο υποσύνολο του συνόλου δεδομένων αυτού, χρησιμοποιώντας το εργαλείο Segments.ai—μια αποτελεσματική πλατφόρμα δεδομένων εκπαίδευσης για μηχανικούς όρασης υπολογιστών, που προσφέρει μια ισχυρή διεπαφή για την επισήμανση δεδομένων. Αποφασίσαμε να επισημάνουμε 420 εικόνες 512x512 (που αντιστοιχούν σε 12 πλήρεις εικόνες 3200x2400) χρησιμοποιώντας το ίδιο σύνολο ετικετών με το σύνολο δεδομένων coastTrain, το οποίο απεικονίζεται στον Πίνακα 3.2 παραπάνω. Έτσι, αποκτήσαμε ένα επισημασμένο σύνολο δεδομένων, το οποίο ονομάζουμε grCoastline, το οποίο θα αξιοποιήσουμε για το τελικό fine-tuning των μοντέλων μας.

Figure 3.9: *Παράδειγμα αρχικής εικόνας με την αντίστοιχη μάσκα (μαύρο: unknown, μπλε: water, άσπρο: whitewater, κίτρινο: sediment, πράσινο: vegetation).*

3.2 Αρχιτεκτονική

3.2.1 Εισαγωγή

Το αρχιτεκτονικό πλαίσιο που χρησιμοποιείται σε αυτή τη διατριβή στοχεύει στην επίτευξη σημασιολογικής κατάτμησης σε παράκτιες εικόνες του συνόλου δεδομένων της Ελληνικής Ακτογραμμής που περιγράφεται στο κεφ. 3.1.2. Η σημασιολογική κατάτμηση στις παράκτιες εικόνες έχει σημαντική σημασία για διάφορες εφαρμογές όπως η παρακολούθηση του περιβάλλοντος, η διαχείριση των ακτών και η αντιμετώπιση καταστροφών. Ο στόχος ήταν να χρησιμοποιηθούν μοντέλα μετασχηματιστών τελευταίας τεχνολογίας, συγκεκριμένα SegFormer, MaskFormer και Mask2Former, για να ταξινομηθεί κάθε pixel στις εικόνες σε μία

από τις επτά διαφορετικές κατηγορίες: water, whitewater, sediment, vegetation, development, other natural terrain, και unknown. Ωστόσο, η σημαντική πρόκληση ήταν ότι το σύνολο δεδομένων της ελληνικής ακτογραμμής ήταν unlabeled, καθιστώντας το ακατάλληλο για άμεση εκπαίδευση. Για την αντιμετώπιση αυτής της πρόκλησης, ακολουθήθηκαν τρεις διαφορετικές προσεγγίσεις, η καθεμία με στόχο την προσαρμογή των μοντέλων στα χαρακτηριστικά του συνόλου δεδομένων της Ελληνικής Ακτογραμμής. Στο παρακάτω σχήμα 3.10 παρουσιάζεται η συνολική αρχιτεκτονική της τελικής διαδικασίας εκπαίδευσης που ακολουθήσαμε. Οι επιμέρους διαδικασίες περιγράφονται στη συνέχεια.

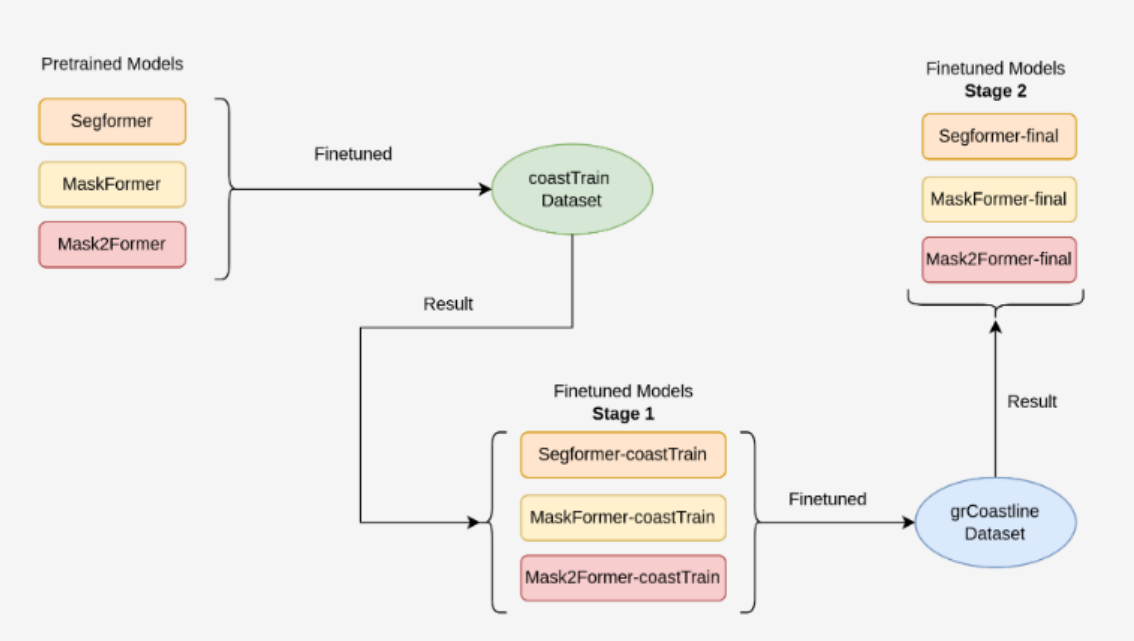


Figure 3.10: *Συνολική Αρχιτεκτονική*

3.2.2 Εκπαίδευση - Στάδιο 1

Η πρώτη προσέγγιση περιελάμβανε την αναζήτηση για ένα προεπισημασμένο σύνολο δεδομένων παράκτιων περιβαλλόντων που μοιράζονται παρόμοια χαρακτηριστικά με το δικό μας σύνολο δεδομένων και περιέχουν ετικέτες σχετικές με την εργασία τμηματοποίησης. Σε αυτήν την αναζήτηση, βρέθηκε το σύνολο δεδομένων coastTrain [25]. Υποβλήθηκε, στην πορεία, σε προεπεξεργασία για να ευθυγραμμιστεί με τις απαιτήσεις εκπαίδευσης, μια διαδικασία που περιγράφεται λεπτομερώς στο Κεφάλαιο 3.1.1. Όσον αφορά τα μοντέλα μετασχηματιστών, αναζητήθηκαν προεκπαιδευμένα μοντέλα προσαρμοσμένα για εργασίες σημασιολογικής κατάτμησης για τη βελτίωση της απόδοσης των μοντέλων μας, σύμφωνα με όσα περιγράφηκαν στο κεφάλαιο για την μεταφορά μάθησης (κεφ. 2.3). Δεδομένης της ανάγκης των μοντέλων μετασχηματιστών για εκπαίδευση σε πολύ μεγάλα σύνολα δεδομένων, εντοπίστηκαν προεκπαιδευμένα μοντέλα που εκπαιδεύτηκαν σε εκτεταμένα σύνολα δεδομένων όπως τα Cityscapes [19] και ADE20K [23]. Αυτά τα μοντέλα στη συνέχεια βελτιστοποιήθηκαν χρησιμοποιώντας το σύνολο δεδομένων coastTrain, με αποτέλεσμα την ανάπτυξη μοντέλων σταδίου 1. Τα μοντέλα σταδίου 1 θα χρησιμοποιηθούν για εξαγωγή συμπερασμάτων στο σύνολο δεδομένων της Ελληνικής Ακτογραμμής για την οπτική αξι-

ολόγηση των αποτελεσμάτων τμηματοποίησης, τα οποία εμφανίζονται στο Κεφάλαιο 3.4. Αυτά τα μοντέλα θα χρησιμεύσουν επίσης ως ραχοκοκαλιά για περαιτέρω εκπαίδευση στο επόμενο στάδιο.

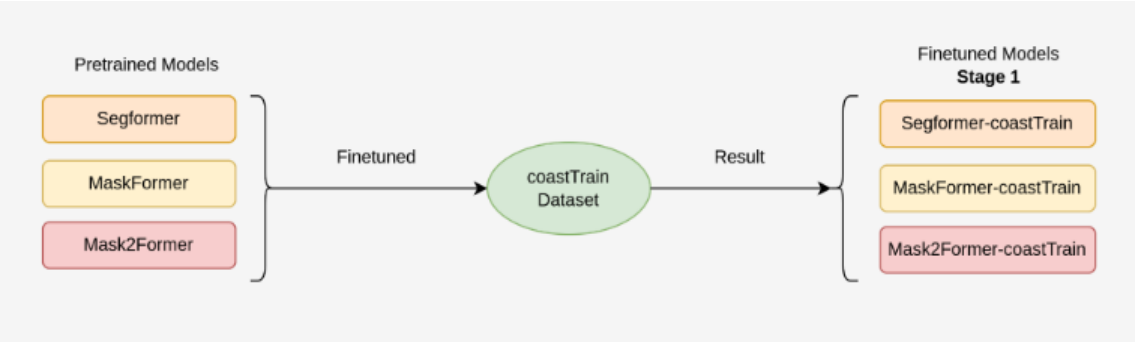


Figure 3.11: *Αρχιτεκτονική - Στάδιο 1*

3.2.3 Εκπαίδευση - Στάδιο 2

Κατά την αξιολόγηση των μοντέλων σταδίου 1 στο σύνολο δεδομένων της Ελληνικής Ακτογραμμής, κατέστη προφανές ότι χρειαζόταν περαιτέρω λεπτομερής ρύθμιση για την προσαρμογή των μοντέλων στα συγκεκριμένα χαρακτηριστικά του ελληνικού συνόλου δεδομένων. Στη δεύτερη προσέγγιση, τα αποτελέσματα των μοντέλων του σταδίου 1 βελτιώθηκαν με περαιτέρω προσαρμογή τους σε ένα υποσύνολο του ελληνικού συνόλου δεδομένων που επισημάναμε χειροκίνητα, όπως περιγράφεται αναλυτικά στο Κεφάλαιο 3.1.2. Αυτό το υποσύνολο (grCoastline) περιέχει τις ίδιες ετικέτες με το σύνολο δεδομένων coastTrain. Έτσι, τα μοντέλα σταδίου 1 έγιναν fine tuned στο σύνολο δεδομένων grCoastline για να προσαρμόσουν τις προβλέψεις των μοντέλων στις δικές μας εικόνες. Αυτή η εκπαιδευτική διαδικασία οδήγησε στη δημιουργία μοντέλων σταδίου 2, τα οποία χρησιμοποιούμε επίσης για να εξετάσουμε οπτικά τα αποτελέσματά τους σε σύγκριση με τα μοντέλα του σταδίου 1.

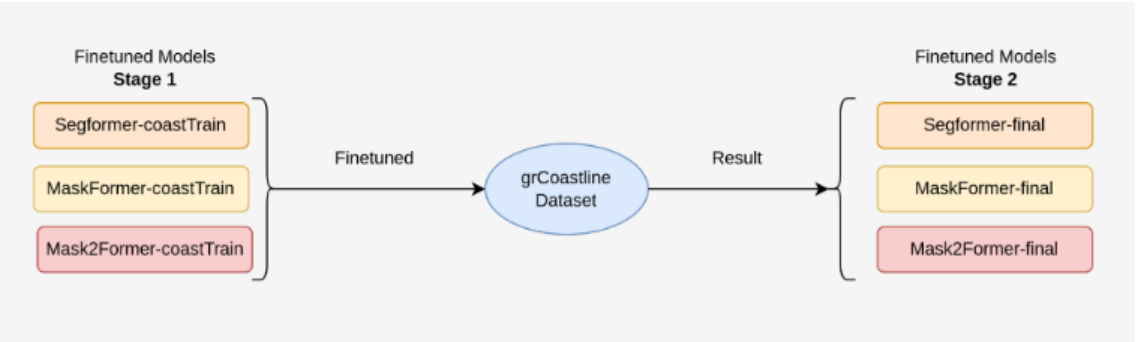


Figure 3.12: *Αρχιτεκτονική - Στάδιο 1*

3.2.4 Απευθείας Εκπαίδευση με το grCoastline dataset

Στην τρίτη προσέγγιση, τα προεκπαιδευμένα μοντέλα προσαρμόστηκαν απευθείας στο σύνολο δεδομένων της Ελληνικής Ακτογραμμής (grCoastline) για λόγους σύγκρισης ώστε να αξιολογηθεί η σημασία της χρήσης του συνόλου δεδομένων coastTrain στην προηγούμενη προσέγγιση. Αυτή η προσέγγιση είχε ως στόχο να καθορίσει εάν το βήμα της αναζήτησης ενός

προεπισημασμένου συνόλου παράκτιων δεδομένων για την πρώτη εκπαίδευση των μοντέλων έπαιξε σημαντικό ρόλο στα αποτελέσματά μας.

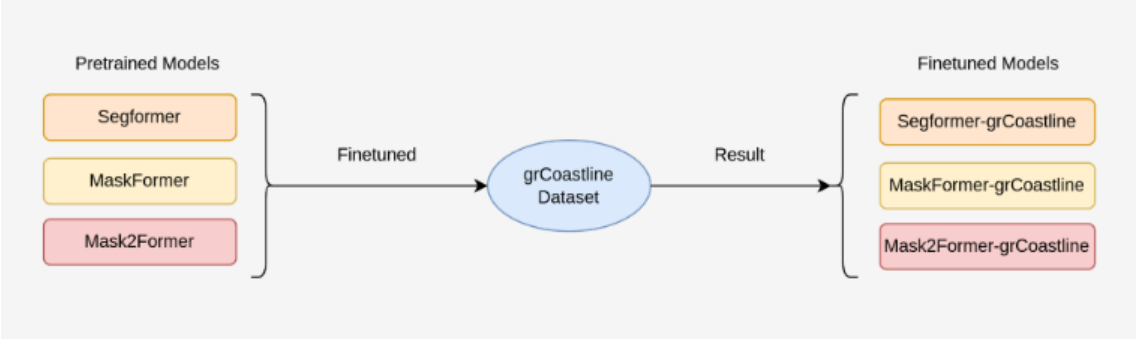


Figure 3.13: *Αρχιτεκτονική απευθείας εκπαίδευσης*

3.3 Αποτελέσματα

Στους παρακάτω πίνακες παρουσιάζουμε τα αποτελέσματα της εκπαιδευτικής διαδικασίας για κάθε στάδιο και για τα τρία μοντέλα. Πιο συγκεκριμένα, παρουσιάζουμε τις τιμές των μετρικών που περιγράφονται στο κεφάλαιο 2.6, αλλά χρησιμοποιούμε το mIoU ως τη κύρια μέτρηση αξιολόγησης που είναι η καταλληλότερη για την εργασία της σημασιολογικής κατάτμησης. Εξετάζουμε διαφορετικές εκδόσεις για κάθε μοντέλο σχετικά με το μέγεθος τους και το σύνολο δεδομένων που χρησιμοποιούνται για την προεκπαίδευση. Για κάθε στάδιο εκπαίδευσης συγκρίνουμε τις διαφορετικές εκδόσεις των μοντέλων και επίσης τα 3 μοντέλα μεταξύ τους.

3.3.1 Στάδιο 1 - coastTrain dataset

SegFormer

Για το SegFormer εκπαιδεύσαμε συνολικά 4 διαφορετικές εκδόσεις του μοντέλου. Όπως αναφέρεται στο [2], υπάρχει μια κλιμακωτή προσέγγιση για το SegFormer που έχει μια σειρά μοντέλων από το SegFormer-B0 έως το SegFormer-B5, επιτυγχάνοντας σημαντικά καλύτερη απόδοση όσο αυξάνεται το μέγεθος του μοντέλου. Για παράδειγμα, το SegFormer-B4 φτάνει το 51,1% mIoU στο ADE20K ενώ το SegFormer-B5 φτάνει το 51,8%. Στο Cityscapes, το SegFormer-B4 φτάνει το 83,8% και το SegFormer-B5 το 84,0%. Επιλέξαμε να βελτιστοποιήσουμε τα SegFormer-B4 και SegFormer-B5 προεκπαιδευμένα τόσο στο Cityscapes όσο και στο ADE20K προκειμένου να αξιολογήσουμε την απόδοσή τους.

Στάδιο 1 SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	79.20	86.09	93.54
SegFormer-B4	ADE	76.15	85.69	92.83
SegFormer-B5	Cityscapes	**82.69**	90.60	94.23
SegFormer-B5	ADE	77.37	85.45	92.45

Table 3.4: *Στάδιο 1 - Αποτελέσματα των SegFormer μοντέλων*

Όπως ήταν αναμενόμενο, το SegFormer-B5 είχε καλύτερη απόδοση από το SegFormer-B4. Επιπλέον, τα μοντέλα που ήταν προεκπαιδευμένα στο Cityscapes πέτυχαν καλύτερα αποτελέσματα από αυτά που προεκπαιδεύτηκαν στο ADE20K, κάτι που είναι επίσης δικαιολογημένο καθώς το ADE20K είναι ένα πιο σύνθετο σύνολο δεδομένων σε σύγκριση με το Cityscapes που ταιριάζει καλύτερα στα δεδομένα του coastTrain. Το καλύτερο μοντέλο SegFormer στο Στάδιο 1 είναι το SegFormer-B5 προεκπαιδευμένο στο Cityscapes και επιτυγχάνει 82,69% mIoU στο σύνολο δεδομένων coastTrain.

MaskFormer

Για το MaskFormer εκπαιδεύσαμε 2 εκδόσεις του μοντέλου σχετικά με το μέγεθος της ραχοκοκαλιάς (backbone) που χρησιμοποιούν, και οι δύο προεκπαιδευμένες στο ADE20K. Το MaskFormer-Base χρησιμοποιεί έναν κωδικοποιητή Swin-B ενώ το MaskFormer-Large έναν Swin-L. Πιο συγκεκριμένα, κάναμε fine tune στο coastTrain τα μοντέλα MaskFormer-Base και MaskFormer-Large, και τα δύο προεκπαιδευμένα στο ADE20K. Σύμφωνα με το [3], το MaskFormer-Base επιτυγχάνει 53,9% mIoU στο ADE20K, ενώ το MaskFormer-Large 55,6% mIoU.

Στάδιο 1 MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE	81.4	89.64	93.96
MaskFormer-Large	ADE	**82.18**	91.76	94.79

Table 3.5: *Στάδιο 1 - Αποτελέσματα των MaskFormer μοντέλων*

Όπως ήταν αναμενόμενο, το μοντέλο MaskFormer-Large προεκπαιδευμένο στο ADE20K απέδωσε καλύτερα του MaskFormer-Base, επιτυγχάνοντας 82,18% mIoU στο coastTrain.

Mask2Former

Για το μοντέλο Mask2Former βελτιστοποιήσαμε αυτό με τη ραχοκοκαλιά Swin-L (δηλαδή Mask2Former-Large). Η πρώτη έκδοση είναι προεκπαιδευμένη στο Cityscpaes ενώ η δεύτερη στο ADE20K. Σύμφωνα με το [4], στο ADE20K το Mask2Former-Large επιτυγχάνει 57,7% mIoU ενώ στο Cityscapes επιτυγχάνει 83,3% mIoU.

Στάδιο 1 Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**84.06**	93.42	95.45
Mask2Former-Large	ADE	83.92	91.14	94.67

Table 3.6: *Στάδιο 1 - Αποτελέσματα των Mask2Former μοντέλων*

Το Mask2Former προεκπαιδευμένο στο Cityscapes είχε καλύτερη απόδοση από αυτό που προεκπαιδεύτηκε στο ADE20K. Αυτό το αποτέλεσμα δικαιολογείται για δύο βασικούς λόγους. Πρώτον, οι βαθμολογίες mIoU των προεκπαιδευμένων μοντέλων διαφέρουν σημαντικά, καθώς το Mask2Former επιτυγχάνει 83,3% mIoU στο Cityscapes και 57,7% mIoU στο ADE20K, κυρίως λόγω της πολυπλοκότητας και του μεγαλύτερου αριθμού κλάσεων που διαθέτει το ADE20K. Δεύτερον, το σύνολο δεδομένων Cityscapes είναι πολύ πιο συμβατό με το σύνολο δεδομένων coastTrain από το ADE20K, καθώς μοιράζεται ορισμένες κλάσεις και επίσης ο τύπος περιεχομένου είναι πιο συμβατός με την εργασία μας.

3.3.2 Απευθείας εκπαίδευση - grCoastline dataset

Συνεχίζοντας με την απευθείας εκπαίδευση με το σύνολο δεδομένων grCoastline, χρησιμοποιήσαμε τα ίδια μοντέλα όπως στο στάδιο 1. Τα αποτελέσματα ακολούθησαν το ίδιο μοτίβο με τα ίδια μοντέλα να έχουν καλύτερη απόδοση. Οι βαθμολογίες εκπαίδευσης για κάθε μοντέλο παρουσιάζονται στους παρακάτω πίνακες.

SegFormer

Απευθείας Εκπαίδευση SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	72.21	77.96	94.29
SegFormer-B4	ADE	72.07	78.64	94.37
SegFormer-B5	Cityscapes	**72.46**	78.87	94.56
SegFormer-B5	ADE	69.51	77.38	93.81

Table 3.7: *Απευθείας εκπαίδευση - Αποτελέσματα των SegFormer μοντέλων*

Το SegFormer-B5 προεκπαιδευμένο σε Cityscapes πέτυχε την υψηλότερη βαθμολογία με 72,46% mIoU.

MaskFormer

Απευθείας Εκπαίδευση MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE	75.55	80.24	93.46
MaskFormer-Large	ADE	**78.79**	84.48	94.27

Table 3.8: *Απευθείας εκπαίδευση - Αποτελέσματα των MaskFormer μοντέλων*

Το MaskFormer-Large προεκπαιδευμένο στο ADE20K πέτυχε την υψηλότερη βαθμολογία με 78,79% mIoU.

Mask2Former

Απευθείας Εκπαίδευση Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**82.42**	91.28	96.2
Mask2Former-Large	ADE	81.91	89.03	93.22

Table 3.9: *Απευθείας εκπαίδευση - Αποτελέσματα των Mask2Former μοντέλων*

Το Mask2Former-Large προεκπαιδευμένο στο Cityscapes πέτυχε την υψηλότερη βαθμολογία με 82,42% mIoU.

3.3.3 Στάδιο 2 - coastTrain και grCoastline

Τέλος, στο Στάδιο 2 χρησιμοποιήσαμε επίσης τις ίδιες εκδόσεις των μοντέλων όπως και στα προηγούμενα στάδια. Τα αποτελέσματα και πάλι, όπως περιμέναμε, ακολούθησαν τις ίδιες αρχές και παρουσιάζονται στους παρακάτω πίνακες.

SegFormer

Στάδιο 2 SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	74.77	83.06	92.47
SegFormer-B4	ADE	75.78	83.13	94.51
SegFormer-B5	Cityscapes	**76.81**	84.19	94.79
SegFormer-B5	ADE	72.61	79.41	94.57

Table 3.10: *Στάδιο 2 - Αποτελέσματα των SegFormer μοντέλων*

Το SegFormer-B5 προεκπαιδευμένο σε Cityscapes πέτυχε την υψηλότερη βαθμολογία με 76,81% mIoU.

MaskFormer

Στάδιο 2 MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE	79.91	86.27	93.46
MaskFormer-Large	ADE	**80.59**	89.93	93.87

Table 3.11: *Στάδιο 2 - Αποτελέσματα των MaskFormer μοντέλων*

Το MaskFormer-Large προεκπαιδευμένο στο ADE20K πέτυχε την υψηλότερη βαθμολογία με 80,59% mIoU.

Mask2Former

Στάδιο 2 Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**85.43**	94.33	96.27
Mask2Former-Large	ADE	83.82	92.56	94.97

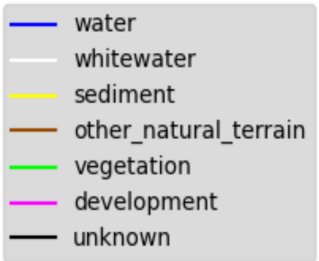
Table 3.12: *Στάδιο 2 - Αποτελέσματα των Mask2Former μοντέλων*

Το Mask2Former-Large προεκπαιδευμένο στο Cityscapes πέτυχε την υψηλότερη βαθμολογία με 85,43% mIoU.

3.4 Οπτικοποίηση Αποτελεσμάτων

Προκειμένου να αξιολογήσουμε και οπτικά τις αποδόσεις των μοντέλων, τα δοκιμάζουμε σε δεδομένα του grCoastline dataset που δεν έχουν χρησιμοποιηθεί για την εκπαίδευσή τους και οπτικοποιούμε τα αποτελέσματά τους.

3.4.1 SegFormer



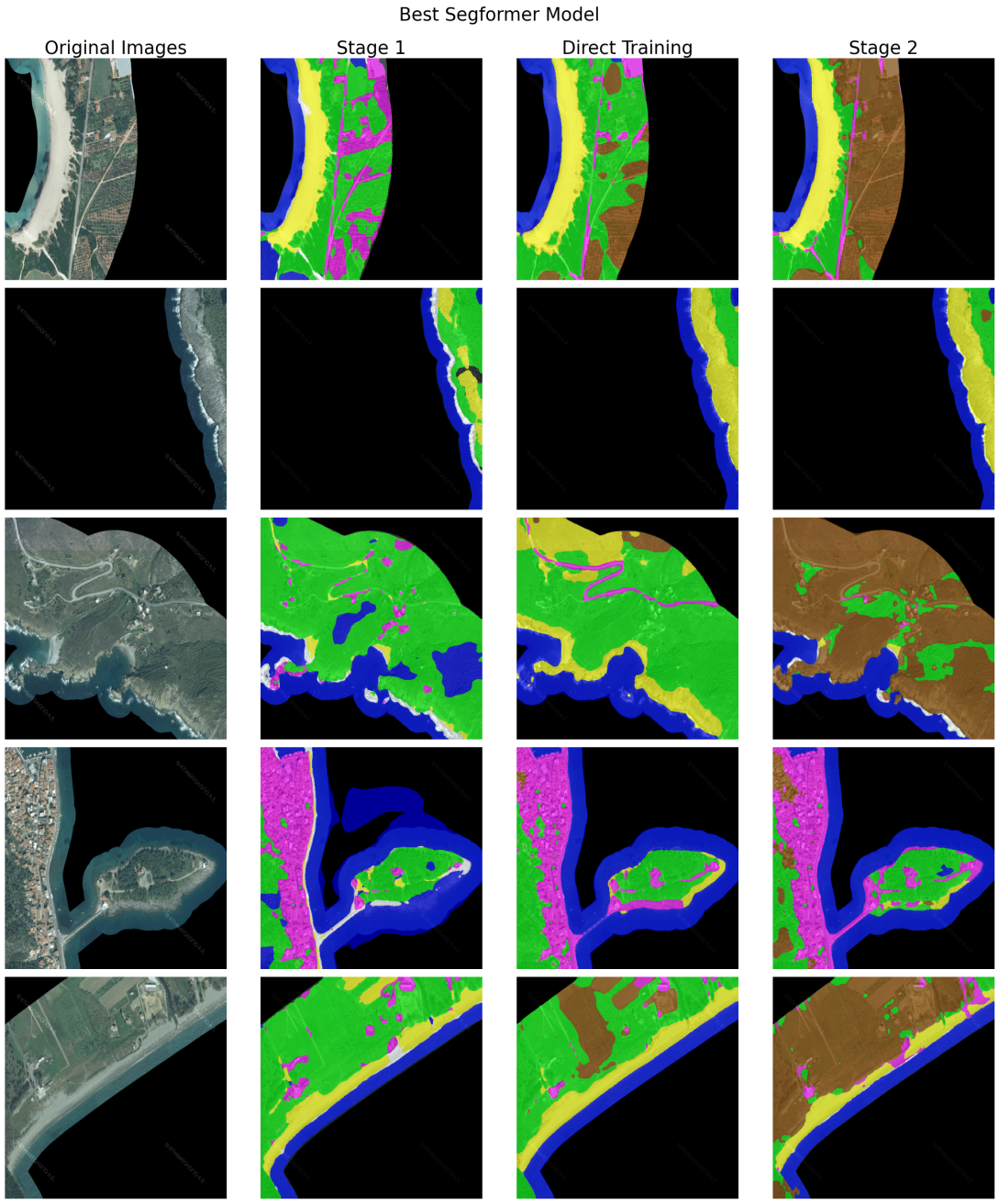


Figure 3.14: *Οπτικοποίηση αποτελεσμάτων του καλύτερου SegFormer μοντέλου*

3.4.2 MaskFormer

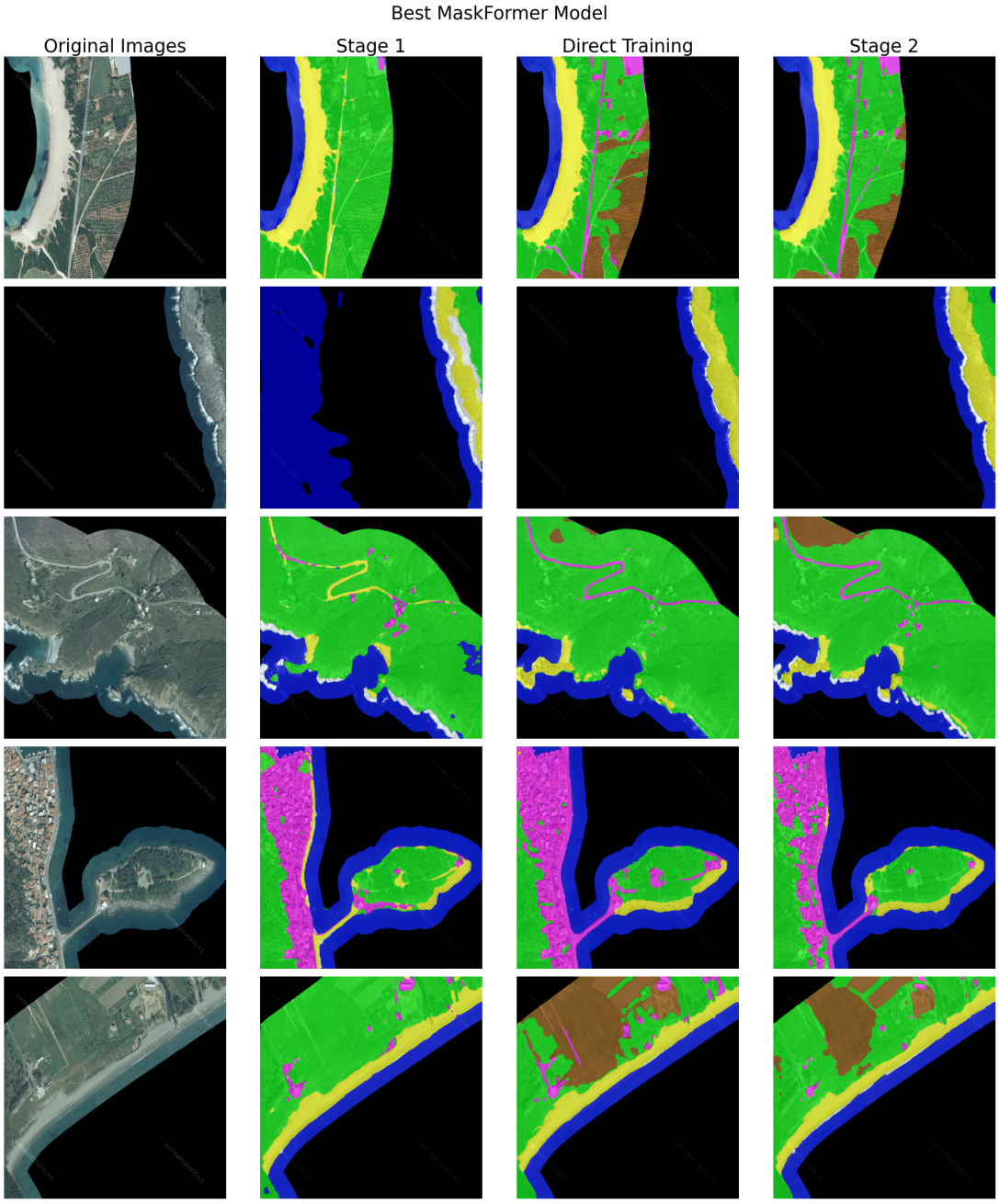


Figure 3.15: *Οπτικοποίηση αποτελεσμάτων του καλύτερου MaskFormer μοντέλου*

3.4.3 Mask2Former

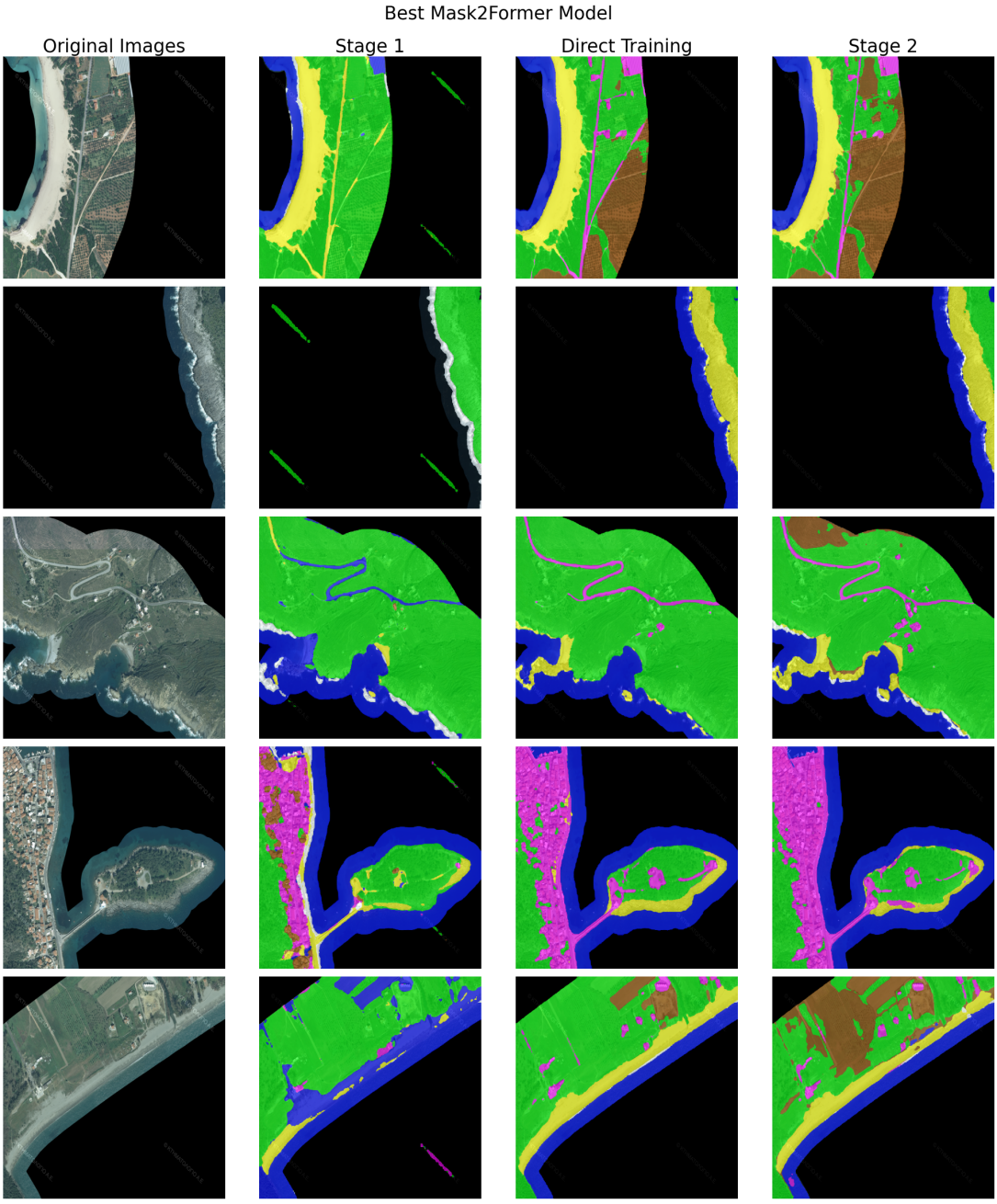


Figure 3.16: *Οπτικοποίηση αποτελεσμάτων του καλύτερου Mask2Former μοντέλου*

3.4.4 Σύγκριση Μοντέλων (Στάδιο 2)

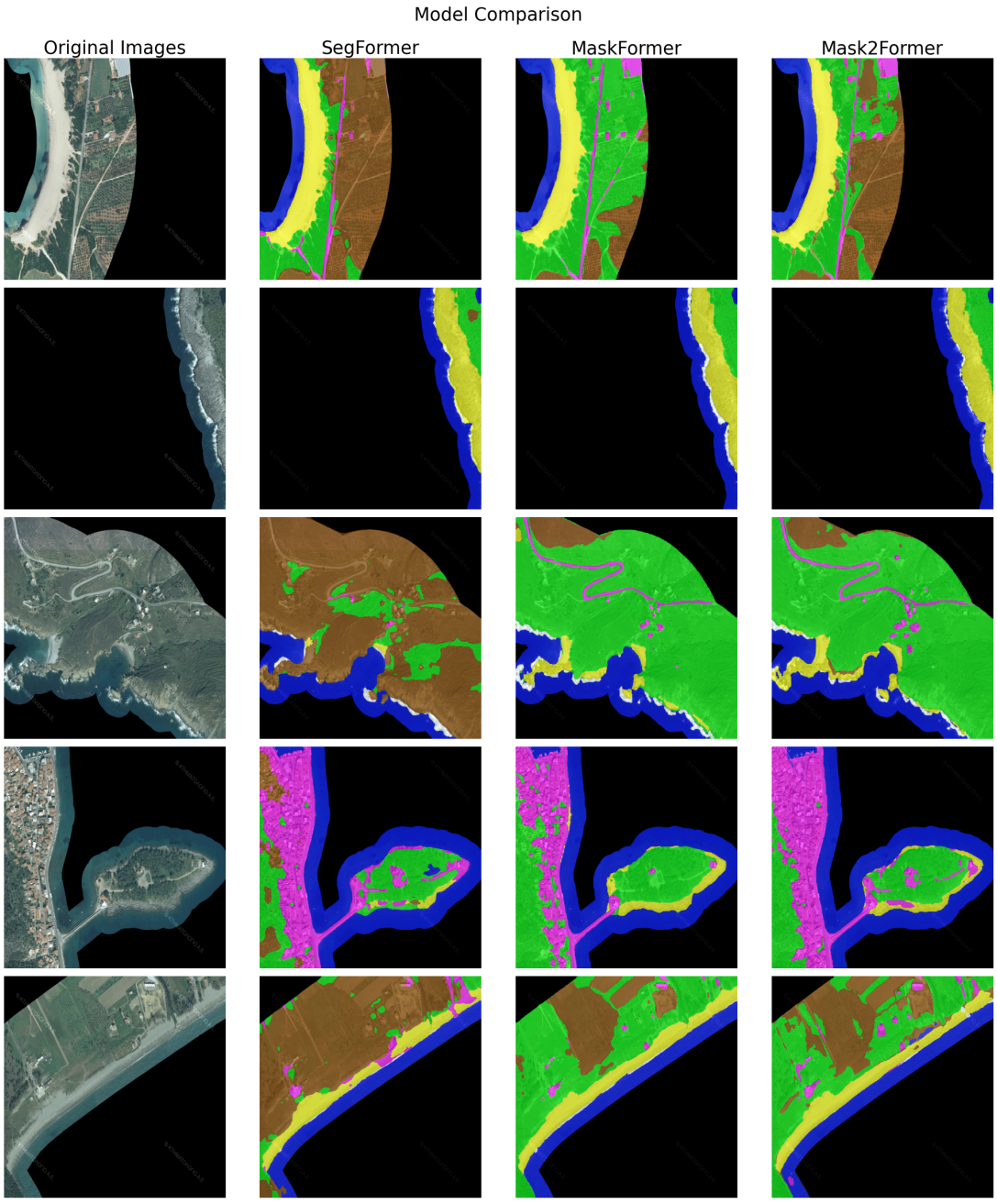


Figure 3.17: *Οπτικοποίηση αποτελεσμάτων των καλύτερων SegFormer, MaskFormer και Mask2Former μοντέλων στο Στάδιο 2*

Chapter 4

Επίλογος

4.1 Συμπεράσματα

Στην παρούσα διπλωματική εργασία καταφέραμε να πραγματοποιήσουμε σημασιολογική κατάτμηση σε δεδομένα τηλεπισκόπησης της Ελληνικής ακτογραμμής με την χρήση μοντέλων Μετασχηματιστών. Με την διαδικασία αυτή εισάγουμε χρήσιμη πληροφρία στις παράκτιες εικόνες σχετικά με το περιεχόμενό τους, εξελίσοοντας έτσι την γενικότερη έρυνα στην απομακρυσμένη συλλογή δεδομένων στις παράκτιες περιοχές. Μέσω της διαδικασίας εκπαίδευσης των μοντέλων αλλά και της οπτικής αξιολόγησής τους, καταλήξαμε ότι το Mask2Former είχε την καλύτερη επίδοση, πετυχαίνοντας 85.43% mIoU στο grCoastline dataset. Επιπλέον, αναδείξαμε την αξία της μεταφοράς μάθησης, καθώς επιλέξαμε να προεκπαιδεύσουμε τα μοντέλα σε ένα επισημασμένο σύνολο δεδομένων με εικόνες από τις ακτές τις Αμερικής και στη συνέχεια να τα προσαρμόσουμε στο δικό μας σύνολο δεδομένων. Συγκρίνοντας τα αποτελέσματα της διαδικασίας αυτής σε σχέση με την απευθείας εκπαίδευση των μοντέλων στο Ελληνικό dataset, η υπεροχή της αξιοποίησης της μεταφοράς μάθησης είναι εμφανής. Συνολικά, κάναμε ένα αρχικό, αλλά ζωτικής σημασίας, βήμα προς την κατεύθυνση της αξιοποίησης των τεχνικών υπολογιστικής όρασης για να βοηθήσουμε περαιτέρω τη τρέχουσα in-situ έρευνα. Σε αυτή τη διατριβή, εξετάσαμε και προεπεξεργαστήκαμε σύνολα δεδομένων εικόνων, αναζητήσαμε και μελετήσαμε μοντέλα SOTA στο έργο της σημασιολογικής κατάτμησης και τα εκπαιδεύσαμε επιτυχώς, αποκτώντας σημαντικά αποτελέσματα.

4.2 Μελλοντικές Επεκτάσεις

Όπως αναφέρεται και παραπάνω, η παρούσα εργασία αποτελεί το πρώτο και βασικό βήμα στα πλαίσια μίας γενικότερης έρευνας που αφορά στην γεωμορφολογική ανάλυση παράκτιων περιβάλλοντων μέσω δεδομένων τηλεπισκόπησης. Κάποια από τα μελλοντικά βήματα της έρευνας αυτής είναι:

1. Εισαγωγή περισσότερων κλάσεων για την πιο λεπτομερή σημασιολογική κατάτμηση των παράκτιων εικόνων. Για παράδειγμα, οι ακτές θα μπροσούσαν να διαχωρίζονται σε αμμώδεις ή βραχώδεις, ενώ οι περιοχές ανάπτυξης σε σπίτια, δρόμους, λιμάνια κλπ. Οι περισσότερες αυτές κλάσεις θα προσέφεραν χρήσιμη και πιο καθολική πληροφορία για το περιεχόμενο των παράκτιων περιοχών.

2. Αξιοποίηση της τρίτης διάστασης που συνοδεύει τις εικόνες του Ελληνικού Κτηματολογίου, αυτή του ύψους που βρίσκεται το κάθε pixel σε σχέση με το επίπεδο της θάλασσας. Η πληροφορία αυτή του ύψους, θα μπορούσε είναι ιδιαίτερα χρήσιμη για την ανάλυση της κλίσης σε ορισμένες περιοχές.

3. Οι αεροφωτογραφίες του Κτηματολογίου έχουν ανάλυση 25cm, η οποία θεωρείται ιδιαίτερα υψηλή και μπορεί να χρησιμοποιηθεί και για ανάλυση σε επίπεδο υλικού. Ειδικότερα, υπάρχει η ανάγκη εξαγωγής πληροφορίας σχετικά με το υλικό των παραλιών (είδος άμμου, χαλικιού κλπ), η οποία μέχρι σήμερα πραγματοποιείται με in-situ μετρήσεις.

Παράλληλα, για την βελτιστοποίηση των αποτελεσμάτων μας μπορούν επίσης να αξιοποιηθούν τα ακόλουθα:

1. Βελτιστοποίηση των υπερπαραμέτρων εκπαίδευσης των μοντέλων (learning rate, batch size κλπ.).

2. Χρήση περισσότερων επισημασμένων δεδομένων για την εκπαίδευση. Ως πηγή θα μπορούσε να χρησιμοποιηθεί το Open Street Map.

3. Αξιοποίηση του 4ου band των φωτογραφιών (υπέρυθρο) το οποίο μπορεί επίσης να προσφέρει χρήσιμη πληροφορία.

4. Εκπαίδευση του μοντέλου OneFormer το οποίο αποτελεί το νέο SOTA για τις εργασίες κατάτμησης εικόνας.

Part II

English Version

Chapter 5

Introduction

5.1 Motive

Coastal areas represent dynamic and ecologically diverse areas that are vital for various socio-economic activities, environmental sustainability and biodiversity conservation. Understanding and monitoring these complex ecosystems requires efficient tools and methodologies capable of extracting detailed information from large and diverse datasets. Remote sensing, particularly through the use of aerial and satellite imagery, has emerged as a key technology for this problem, offering significant potential for comprehensive analysis of coastlines [27, 28, 29]. Semantic segmentation is a fundamental task in computer vision through which meaningful information is extracted from images by separating them into semantically meaningful regions. At the same time, recent developments in Transformer-based models have revolutionized the field of computer vision, offering powerful alternatives to traditional convolutional neural networks (CNNs). Vision transformer models have shown remarkable performance in capturing long-range dependencies and information, making them suitable for analyzing coastal imagery. The work was motivated by the provision of a set of high-resolution aerial images of the Greek coastline by the Hellenic Land Registry. In collaboration with the Department of Geology and Geoenvironment of EKPA, we studied the specific dataset and concluded that its appropriate analysis and processing can offer particularly important information for the geomorphological recording of the coastal environment of Greece and strengthen remote sensing in an effort to limit the difficulties of in-situ missions.

5.2 Scope of Work

This diploma thesis constitutes the first but particularly pivotal step of this research, which is the application of state-of-the-art transformer models for the semantic segmentation of coastal images. More specifically, the detection of distinct land surface classes such as water bodies, vegetation, sediments and other physical and non-physical features at the pixel level. The main problem was that these images do not contain any kind of labeling so they cannot be used to train the models directly. For this reason our research focuses on leveraging pre-existing labeled datasets from the US coastline for the training of transformer models and their further adaptation to the specific characteristics of the

Greek coastline. The methodology involves initial model training on a labeled dataset sourced from multiple sources, including orthophotos and satellite imagery, followed by inference on an unlabeled dataset of the Greek coastline. Subsequently, we undertake manual labeling of a subset of the Greek dataset to refine and adapt the models to the unique coastal features of the study area. Through rigorous evaluation and comparison of model performance, we seek to ascertain the effectiveness of different transformer architectures in addressing the challenges of semantic segmentation in coastal environments. Our goal is to evaluate the effectiveness of the SegFormer, MaskFormer, and Mask2Former models in accurately delineating various land surface classes. It should be mentioned that many deep neural architectures have been developed and applied for segmentation and prediction in a variety of applications by members of the Artificial Intelligence and Learning Systems Lab of the National Technical University of Athens. Bayesian models with capsules and uncertainty estimation, semi and self-supervised learning algorithms, domain adaptation, augmentation, transformers and attention methodologies have been developed and used in applications, such as medical imaging [30, 31], image captioning [32], fault detection in nuclear power stations [33, 34], agri-food production prediction [35, 36], human behavior prediction [37, 38], as well as for capsule networks [39, 40, 41] and transparency [42, 43].

5.3 Structure

The first part [6] covers the Theoretical Background in which we delve into the theoretical foundations of remote sensing, image segmentation, transfer learning and transformer-based models. The second part [7] covers the Experimental part in which we describe in detail the methodology used for the acquisition and preprocessing of the data. Then, the training of the models and their evaluation through the experimental results are presented. Finally, in [8] we provide the extraction of the final conclusions and present possible future directions.

Chapter 6

Theoretical Part

6.1 Remote Sensing

Remote sensing is defined, according to [44], as the measurement of object properties on the earth's surface using data acquired from aircraft and satellites. It is a powerful and versatile technology that plays a crucial role in environmental monitoring, providing valuable insights into Earth's dynamic systems. This non-invasive technique involves the collection and interpretation of information about the Earth's surface from a distance, relying on the interaction of electromagnetic radiation with the Earth's surface. Different objects and materials reflect, emit and absorb energy at distinct wavelengths, allowing sensors to capture information about their composition and characteristics. The electromagnetic spectrum, which includes visible, infrared, and microwave wavelengths, is utilized to gather data across a wide range of environmental features. In this work, we will limit the discussion to remote sensing of the earth's surface using optical signals, which means that our analysis will be performed over a two-dimensional spatial grid, i.e., images. The successful analysis of these satellite or aerial images can be a cost-effective alternative to in situ measurements and it can also help limit the difficulties that arise from field assessments. This imagery analysis can be performed using Machine learning techniques, and more specifically Neural Networks, which is also the subject of this work. Recent advancements have led to the availability of high-resolution satellite imagery, enabling detailed mapping and monitoring of smaller environmental features. This is particularly beneficial for urban planning, precision agriculture, and habitat assessment. Hyperspectral sensors capture data across a multitude of narrow spectral bands, allowing for detailed analysis of materials and vegetation. This technology enhances the discrimination of subtle environmental changes and improves the accuracy of resource assessments. Integration of machine learning algorithms with remote sensing data enhances the automation of image analysis and classification. Additionally, data fusion techniques combine information from multiple sensors to provide a more comprehensive understanding of the environment.

6.2 Image Segmentation

Image segmentation is a crucial task in computer vision that involves partitioning an image into a set of non-overlapping regions whose union is the entire image [15]. This identification of regions, enables a more advanced analysis and understanding of visual content. The resulting regions should be homogeneous and have significantly different values in regards to some characteristic. They should also have boundaries that are smooth and do not contain many holes. In image segmentation, an image has two main components: things and stuff. Things correspond to countable objects in an image (e.g., people, flowers, birds, animals, etc.). In comparison, stuff represents amorphous regions (or repeating patterns) of similar material, which is uncountable (e.g., road, sky, and grass). The task of image segmentation has been dealt with various image processing techniques throughout the years. Some of those initial techniques involve clustering, used with edge and contour detection [16], and histogram approaches like HOG [17] and SIFT [18] feature extraction. However, the introduction of Convolutional Neural Networks (CNNs) contributed a lot in the evolution of image segmentation which took a turn into supervised learning, achieving much better results. A lot of segmentation datasets have grown extensively since then, like Cityscapes [19], ImageNet [20] and COCO [21]. Overall, image segmentation is a quite complex computer vision task and more demanding than image classification as it requires pixel-level classification and relationship detection across various scales, thus, it needs more sophisticated model structures that take into account both semantics and location. In that direction, Neural Networks are capable of learning patterns and deep neural networks have numerous parameters that can imitate complex functions, to achieve this task. Transformer models, which we will examine in this work, are the most recent architectures that deal with segmentation tasks, surpassing the performance of CNNs. The process of image segmentation plays a vital role in various applications, such as remote sensing, scene understanding, and autonomous systems. Different semantics for grouping pixels, e.g., category or instance membership, have led to different types of segmentation tasks, such as panoptic, instance or semantic segmentation [45].

6.2.1 Semantic Segmentation

Semantic segmentation, which is the focus of this work, is a specific type of image segmentation where the goal is to assign a semantic label to each pixel in an image. More specifically, to segment the input image according to semantic information and predict the semantic category of each pixel from a given label set. The output of semantic segmentation is a pixel-wise segmentation map, where each pixel is assigned a class label representing the type of object or region it belongs to. Semantic segmentation studies the uncountable stuff in an image. It analyzes each image pixel and assigns a unique class label based on the texture it represents. Semantic segmentation was designed to recognize stuff which are formless regions of similar texture or material [22]. It does not distinguish between objects of the same class and rather groups them together. This

approach provides a high-level understanding of the image content by categorizing pixels into predefined classes. Starting from Fully Convolutional Networks (FCNs), most deep learning-based semantic segmentation approaches formulate semantic segmentation as per-pixel classification, applying a classification loss to each output pixel. Per-pixel predictions in this formulation naturally partition an image into regions of different classes. Semantic segmentation can be seen as an extension of image classification from image level to pixel level. Researchers focused on improving FCN from different aspects such as: enlarging the receptive field; refining the contextual information; introducing boundary information; designing various attention modules; or using AutoML technologies. These methods significantly improve semantic segmentation performance at the expense of introducing many empirical modules, making the resulting framework computationally demanding and complicated. More recent methods have proved the effectiveness of Transformer-based architectures for semantic segmentation.

Figure 6.1: *Semantic segmentation examples [5]*

6.2.2 Instance Segmentation

Instance Segmentation is a computer vision task that involves identifying and separating individual objects within an image, including detecting the boundaries of each object and assigning a unique label to each object. It takes the segmentation process a step further by not only assigning a semantic label to each pixel but also distinguishing between different instances of the same class. Instance segmentation typically deals with tasks related to countable things. It can detect each object or instance of a class present in an image and assigns it a different mask or bounding box with a unique identifier. This level of detail is particularly valuable in applications where accurate object counting, tracking, or interaction analysis is essential. The goal of instance segmentation is to produce a pixel-wise segmentation map of the image, where each pixel is assigned to a specific object instance. It is very similar to semantic segmentation except in this task object instances are detected separately contributing to studying things and not classes in general. While seemingly related, the datasets, details, and metrics for these two visual recognition tasks -instance and semantic segmentation- vary substantially.

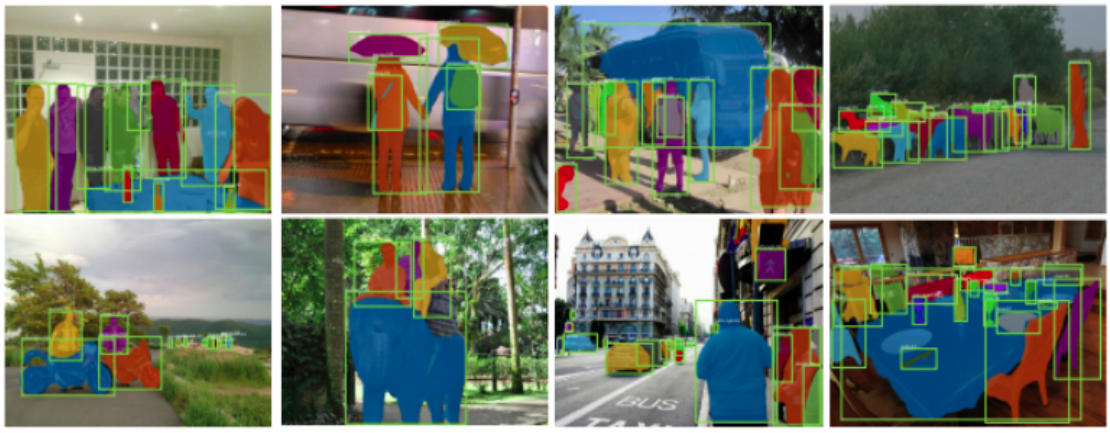


Figure 6.2: *Instance segmentation examples [6]*

6.2.3 Panoptic Segmentation

Panoptic segmentation [46] extends the concept of image segmentation by unifying semantic and instance segmentation. It presents a unified image segmentation approach where each pixel in a scene is assigned a semantic label (due to semantic segmentation) and a unique instance identifier (due to instance segmentation). Panoptic segmentation assigns each pixel only one pair of a semantic label and an instance identifier. However, objects can have overlapping pixels. In this case, panoptic segmentation resolves the discrepancy by favoring the object instance, as the priority is to identify each thing rather than stuff. In panoptic segmentation the output of every pixel i is a semantic label (l_i) and an instance id (z_i) – $(l_i, z_i) \in L \times N$ where $L = L_{th} \cup L_{st}$ and $L_{th} \cap L_{st} = \varnothing$. When a pixel is labeled with $l_i \in L_{st}$ then the corresponding id is irrelevant. However, some pixels may have a special void label. This task thus comes up with a panoptic quality metric that can quantify the performance of the model in this two-factor task. This holistic approach combines the strengths of semantic and instance segmentation, offering a comprehensive understanding of the visual scene.

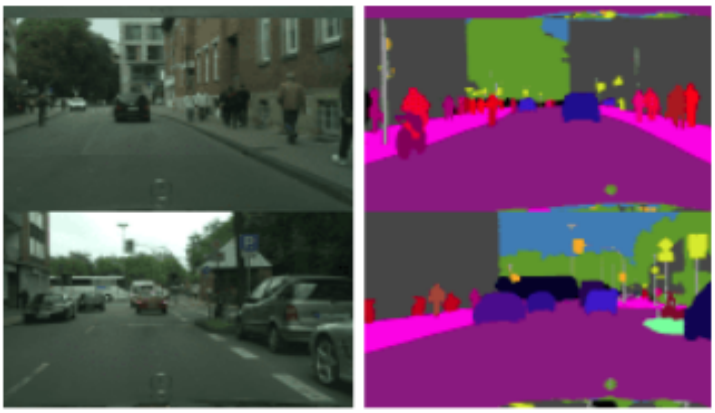


Figure 6.3: *Panoptic segmentation examples [7]*

6.3 Neural Networks

6.3.1 The neuron

Neural networks draw inspiration from the structure of the brain and its fundamental component, the neuron. A neural network consists of layers of neurons, each functioning in a manner that simulates the behavior of biological neurons. The artificial neuron receives one or more inputs and generates an output. These inputs undergo multiplication with adjustable weights, influencing their impact on the final outcome. The output is derived from the weighted sum of inputs, incorporating a modifiable bias and the application of an activation function. This activation function serves to interpret and convey the result in a meaningful manner, such as generating probabilities or binary output. These functions are applied on the dot product of the input signals with the weights of the neuron, introducing a non-linearity to the computation of the result.

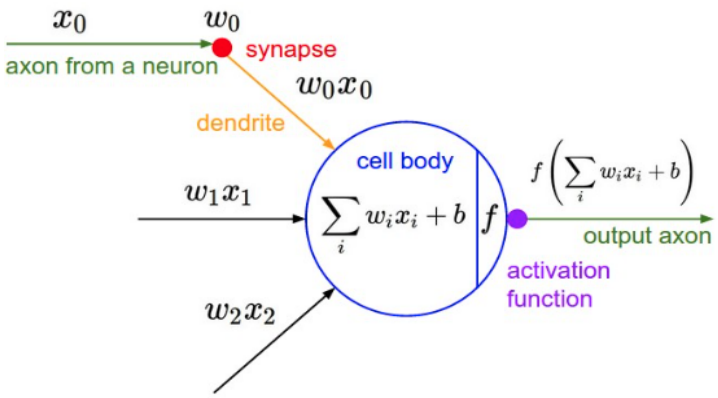


Figure 6.4: *The neuron [8]*

6.3.2 Architecture

Neural Networks [8] consist of neurons with adjustable weights and biases. Each neuron takes in inputs, conducts a dot product, and may apply a non-linearity. The network takes an input (a singular vector) and transforms it through a sequence of hidden layers. Each hidden layer comprises neurons, where each neuron forms a complete connection with all neurons in the preceding layer. Neurons within a single layer operate independently without sharing connections. The final fully-connected layer, termed the "output layer," denotes class scores in classification scenarios. This output is then compared to the desired output or ground truth using a loss function. The loss gradient is utilized to update the weights of the network's layers through the backpropagation algorithm.

Loss function

The loss function [47] evaluates the disparity between the output and the ground truth, and accordingly, the model's hyperparameters are modified during training to minimize this difference. Employing a case-specific loss function, denoted as L, to compute the loss

for a single output against the ground truth, the overall loss is derived by averaging the individual losses across all instances in the training set.

$$J(w^T, b) = \frac{1}{m} \sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{6.1}$$

where w are the weights, b the bias, m is the total number of training set data points, $\hat{y}$ is the prediction and y is the ground truth.

Optimization - Gradient Descent

The loss function enables the quantification of the effectiveness of a specific set of weights, denoted as W. The objective of optimization is to identify W that minimizes the loss function. It is feasible to calculate the optimal direction in which the weight vector should be adjusted, ensuring it is the direction of the steepest descent. This direction is inherently connected to the gradient of the loss function, following this relationship:

$$w^{l+1} = w^l - \eta \frac{\partial J(w^T, b)}{\partial w} \tag{6.2}$$

where η is the learning rate. Gradient Descent is the procedure of repeatedly evaluating the gradient and then performing a parameter update.

Backpropagation Algorithm

Nevertheless, in extensive neural networks, discerning the correlation between certain weights and the loss function proves to be exceedingly challenging. The significant value of the backpropagation algorithm [48] lies in offering a computationally efficient approach to assess intricate derivatives. The inclusion of the fraction in the mean loss calculation is not necessary during the training process, commencing from:

$$J(w) = \sum_{i=1}^{m} L_i(w) \tag{6.3}$$

For a Mean Squared Error function L_i is defined as:

$$L_i = \frac{1}{2} \sum_{k} (\hat{y}_{ik} - y_{ik})^2 \tag{6.4}$$

Which calculates the error when we have a multidimensional output. The input z_i in a unit is transformed into an output a_j of another unit from the dot product with the weight vector of the connection w_{ij}. The sum is then transformed by a non linear activation function h to give the activation z_j of unit j:

$$a_j = \sum_{i} w_{ji} z_i z_j = h(a_j) \tag{6.5}$$

Using the chain rule and 6.5 the derivative of J_i with respect to a weight w_{ij} can be obtained from:

$$\frac{\partial L_i}{\partial w_{ij}} = \frac{\partial L_i}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} \tag{6.6}$$

where we define $\delta_j = \frac{\partial L_i}{\partial a_j}$. For the final output units the gradient of the loss function produces $\delta_k = \hat{y}_k - y_k$ and or the hidden units, δ_k can be calculated from the output units using the chain rule as follows:

$$\delta_j = \frac{\partial L_i}{\partial a_j} = \sum_k \frac{\partial L_i}{\partial a_k} \frac{\partial a_k}{\partial a_j} \delta_j = h'(a_j j) \sum_k w_{kj} \delta_k \tag{6.7}$$

The above steps summarize the backpropagation algorithm. The derivative of the total error J can then be obtained by repeating these steps for each instance of the training set and then by taking the total sum, as:

$$\frac{\partial J}{\partial w_{ij}} = \sum_m \frac{\partial L_m}{\partial w_{ij}} \tag{6.8}$$

Regularization

The objective of a neural network is to acquire a mapping from input to output using training data and subsequently apply it to test data. Therefore, it is crucial for the network to generalize its weights, avoiding specific learning of examples from the training set. When a model perfectly fits the training data, it is termed overfitting. Regularization serves as a method to address the overfitting of Neural Networks. More precisely, it introduces an additional component to the loss function, curbing the excessive increase in weight magnitudes and thereby restraining updates in a less flexible manner.

Dropout

Dropout [49] serves as a training technique aimed at preventing overfitting. The dropout probability signifies the likelihood with which a neuron remains active or is set to zero within the model. It can be conceptualized as randomly sampling a subset of the Neural Network from the complete network and solely adjusting the parameters of the sampled network based on the input data.

Batch size

The batch size stands as a hyperparameter that dictates the number of samples processed before updating the internal model parameters. A batch refers to the set of samples over which a for-loop iterates to make predictions. Following the completion of a batch, predictions are compared to the expected output variables, and an error is computed. The update algorithm is then employed to refine the model, such as descending along the error gradient. A training dataset can be segmented into one or more batches. If an entire training set is utilized to form one batch, the learning algorithm adopts the name batch gradient descent. In cases where the batch size is one sample, the learning algorithm is termed stochastic gradient descent. For batch sizes greater than one sample but less

than the size of the training dataset, the learning algorithm is referred to as mini-batch gradient descent.

6.3.3 Convolution

Convolution [50] is a mathematical operation defined by the following formula:

$$(f * w)[x_1, x_2] = \sum_{m=-k}^{k} f(i,j) \cdot w(x_1 - i, x_2 - j) \tag{6.9}$$

Discrete 2-D convolution involves two matrices with the first argument (f) commonly referred to as the input and the second argument (w) as the kernel. The resultant output is often termed the feature map. This operation plays a pivotal role in Convolutional Neural Networks (CNNs) for image processing and the identification of specific features. In contrast to a conventional Neural Network, CNN layers organize neurons in three dimensions: width, height, and depth. Neurons within a layer are connected to a limited region of the preceding layer, departing from the fully-connected arrangement. As elucidated earlier, a basic CNN comprises sequential layers, each transforming one activation volume into another through a differentiable function. Key layers in a CNN architecture encompass the Convolutional Layer, Pooling Layer, and Fully-Connected Layer. Given the substantial number of pixels in an image, each neuron possesses a receptive field and operates solely on corresponding layers. Consequently, a neuron entails $w \times h \times c$ weights, where $w \times h$ represents the receptive field, and c denotes the number of channels in the input image. This characteristic underscores that convolutional neural networks may fall short in capturing contextual information without attention mechanisms. The depth of the filter, signifies the application of multiple neurons to the same patch of the input image. Consequently, the output image is also three-dimensional, mirroring the depth of the filter. The incorporation of multiple stacked filters aims to identify various characteristics within a single convolutional layer. While each neuron has a designated receptive field, multiple neurons are not required for every patch of the input image. Neurons can be efficiently shifted to compute the output for each patch using the same filter. The shifting is determined by the stride of the filter, indicating the number of pixels the filter traverses across the image before reapplication. Lastly, zero-padding emerges as a critical hyperparameter in a convolutional layer, enabling control over the output dimensions. Assuming the input image is square (disregarding channels) with one dimension W, the convolutional layer having a dimension of F, a stride of S, and using padding P will yield a square output dimension calculated as $\frac{W-F+2P}{S+1}$.

6.4 Transfer Learning

The fundamental concept behind transfer learning is to apply knowledge gained from tasks with sufficient labeled data to situations where only limited labeled data is available. Generating labeled data can be costly, so effectively utilizing existing datasets becomes crucial. Transfer learning is commonly applied in both Convolutional Networks

and Transformers, which we will examine later on, as these models need large datasets for training. It is common to pretrain a ConvNet on a very large dataset (e.g. ImageNet, which contains 1.2 million images with 1000 categories), and then use the ConvNet either as an initialization or a fixed feature extractor for the task of interest, according to [8]. The same also applies for Transformer models which also take advantage of the Transfer learning technique. This procedure has three major variations which depend on the need of each training task and are described bellow:

1. **Fixed feature extractor.** Take a model pretrained on ImageNet, remove the last fully-connected layer, then treat the rest of the model as a fixed feature extractor for the new dataset.

2. **Fine-tuning.** The second strategy is to not only replace and retrain the classifier on top of the model on the new dataset, but to also fine-tune the weights of the pretrained network by continuing the backpropagation. This is motivated by the observation that the earlier features of a model contain more generic features that should be useful to many tasks, but later layers of the model become progressively more specific to the details of the classes contained in the original dataset.

3. **Pretrained models.** Since modern models take a lot of time to train across multiple GPUs on ImageNet, it is common to use trained model checkpoints for fine-tuning.

In traditional machine learning models, the primary aim is to generalize to unseen data based on patterns learned during training. In transfer learning, the goal is to jump-start this generalization process by commencing with patterns learned for a different task. Rather than initiating the learning process from a blank slate, as is often the case with randomly initialized models, transfer learning begins with patterns previously acquired to address a distinct task. Transfer learning is indispensable in the realm of learning, mirroring how humans don't need to be explicitly taught every task to succeed. Individuals frequently encounter novel situations, yet they adapt and solve problems on-the-fly. Transfer learning, therefore, mirrors the ability to glean insights from various experiences and apply that 'knowledge' in novel environments. Particularly in supervised learning, acquiring a substantial amount of labeled data can be prohibitively expensive. The transfer of knowledge is only feasible when it is 'appropriate', but precisely defining appropriateness in this context is challenging, often necessitating experimentation. Transfer learning requires the capability to transfer knowledge from one domain to another.

6.5 CNNs in Image Segmentation

Fully Convolutional Neural Networks (FCN)

Fully Convolutional Networks (FCNs), as explained in [51], are neural networks designed to handle inputs of arbitrary sizes and generate outputs of the same dimensions. These networks exclusively consist of convolutional layers and mechanisms for up-sampling and down-sampling. FCNs aim to adapt state-of-the-art classification Convolutional Neural Networks (CNNs) for segmentation tasks while leveraging their pre-trained weights and

enabling end-to-end pixelwise training. This adaptation involves substituting fully connected layers with 1×1 convolutional layers to maintain output dimensions. The original input size, which may have been reduced by down-sampling in preceding convolutional layers, can be preserved using up-sampling mechanisms. The architecture involves backward convolution with a stride of $\frac{1}{f}$ and learnable weights, facilitating end-to-end learning. Furthermore, to enhance prediction granularity while considering the limitations of down-sampling in output detail, a skip architecture is introduced. This architecture introduces non-linearity by incorporating element-wise addition of predictions from previous layers to up-sampled predictions.

U-Net

Expanding upon the FCN architecture, the U-Net architecture [52] incorporates the skip architecture and transpose convolutions, offering faster training and more precise results. The structure of U-Net is akin to the letter U, symmetrically constructing both the encoder and the decoder. Notably, an enhancement for improved precision involves having a substantial number of features on the decoder side. In contrast to FCN, where the "decoder" side maintains a constant number of features equal to the number of classes, U-Net introduces flexibility with a larger number of features on the decoder side. The encoder comprises three sets of two consecutive convolution layers followed by ReLUs and a pooling layer. This structure is mirrored on the decoder side, with pooling layers replaced by transpose convolutions. Furthermore, an addition module precedes every layer on the decoder side, implementing the skip architecture. This modification contributes to the architecture's precision and is conducive to more accurate segmentation results.

DeepLab

Deeplab [53] builds upon prior architectures by incorporating atrous convolution into FCN structures, replacing deconvolution. Atrous convolution in 1D, involves convolving with a filter that has (r-1) zeroes between its values when the rate of dilation is r. This technique enables a larger receptive field while maintaining the same number of parameters and resolution when used with padding. Deeplab also integrates bilinear interpolation with a fully connected Conditional Random Field (CRF) [54] to transition from an 8x output stride to the original resolution, yielding more detailed results. Additionally, Atrous Spatial Pyramid Pooling (ASPP) is introduced, incorporating atrous convolutions at multiple rates for multi-resolutional processing. Batch normalization is applied after each atrous convolution. Further refinement is seen in Deeplabv3+ [55], which employs the encoder-decoder concept.

HRNet

The genesis of high-resolution convolutional networks stems from the observation that preceding architectures did not maintain high-resolution streams throughout the network. Instead, they often upsampled from low resolution, fused low-level high resolution at the final layers, created medium-resolution streams, or implemented encoder-decoder architectures. HRNet [56] introduced the concept of parallel multi-resolution networks that converge at the end of each stage. The HRNet is structured with four stages, where

each stage integrates one lower resolution in parallel. Within each stage, there are four residual units at every resolution, with each unit comprising two 3×3 convolutions, batch normalization, and ReLU activation. In HRNetV2, the semantic segmentation result is obtained by fusing the four streams with the first stream after upsampling the last three and estimating segmentation maps. This innovative approach ensures the preservation of high resolution across multiple parallel streams, enhancing the network's performance in semantic segmentation tasks.

6.6 Introduction to Transformers

6.6.1 The Transformer Model

The transformer is a simple network architecture that relies entirely on an attention mechanism to draw global dependencies between input and output [1]. Transformers are designed in order to avoid recurrence, which faces memory constraints at longer sequence lengths, not being able to handle them as limitations in batching across examples appear. Attention mechanisms allow modeling of dependencies without regard to their distance in the input or output sequences. Self-attention is an attention mechanism that relates different positions of a single sequence in order to compute a representation of a sequence. Transformer was the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution.

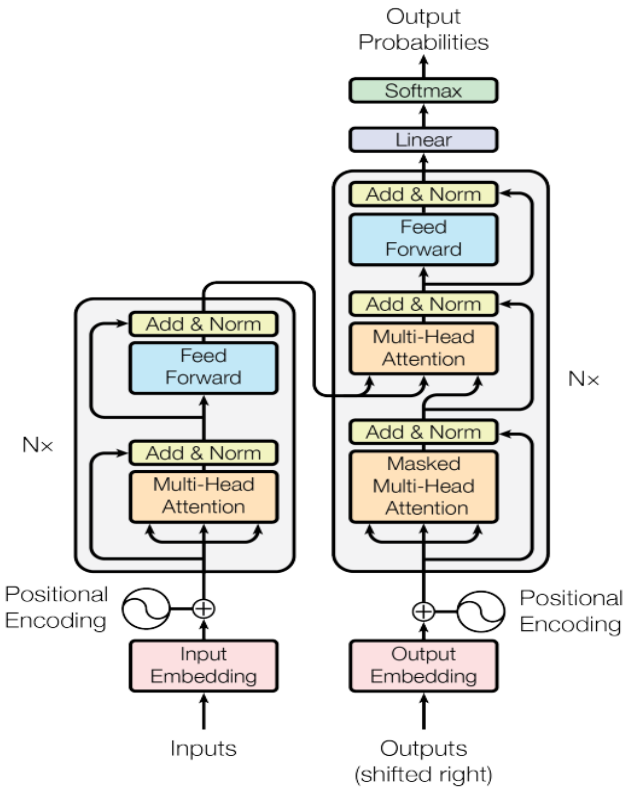


Figure 6.5: *Transformer architecture [1]*

The architecture of a transformer is built upon a sequence of encoders and decoders. The encoder initially transforms the input sequence into a vectorized representation, known as embedding. Specifically, it maps a sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. This sequence is then fed into the decoder, which generates an output sequence of symbols $(y_1, ..., y_n)$ one element at a time. The model is auto-regressive, utilizing previously generated symbols as additional input when generating the next one. In each step, the decoder takes as inputs the embeddings of the input sequence and the embeddings of the previous output, shifted by one token, predicting the next token in the sequence.

Encoder: The encoder consists of a stack of $N = 6$ identical layers, each comprising two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-wise fully connected feed-forward network. Each sub-layer is followed by a residual connection and layer normalization. In other words, the output of each sub-layer is given by $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ represents the function implemented by the sub-layer. To support these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{model} = 512$.

Decoder: The decoder, also composed of a stack of $N = 6$ identical layers, introduces a third sub-layer in addition to the two sub-layers found in each encoder layer. This additional sub-layer performs multi-head attention over the output of the encoder stack. Similar to the encoder, the decoder incorporates residual connections around each sub-layer, followed by layer normalization. The self-attention sub-layer in the decoder stack is modified to prevent positions from attending to subsequent positions. This masking, combined with the offset of output embeddings by one position, ensures that predictions for position i depend only on known outputs at positions less than i.

Attention: The attention function is described as mapping a query and a set of key-value pairs to an output, with all components represented as vectors. The output is computed as a weighted sum of values, where the weight assigned to each value is determined by a compatibility function of the query with the corresponding key.

Query (Q): The query represents the element in the input sequence for which we want to calculate its significance or the relationship of dependency with respect to the other objects. In the context of the attention mechanism, there is usually a query vector associated with each position in the input sequence.

Key (K): The key represents the elements in the input sequence with which the query is compared. Key vectors are used to determine how well each element of the input sequence correlates with the query. Similar to queries, there is usually a key vector for each position in the input sequence.

Value (V): The value represents the information related to each element of the input sequence. Value vectors constitute the pieces of information that are weighted and combined based on the attention scores obtained from the query and key vectors. Like queries and keys, there is a value vector associated with each position in the input sequence.

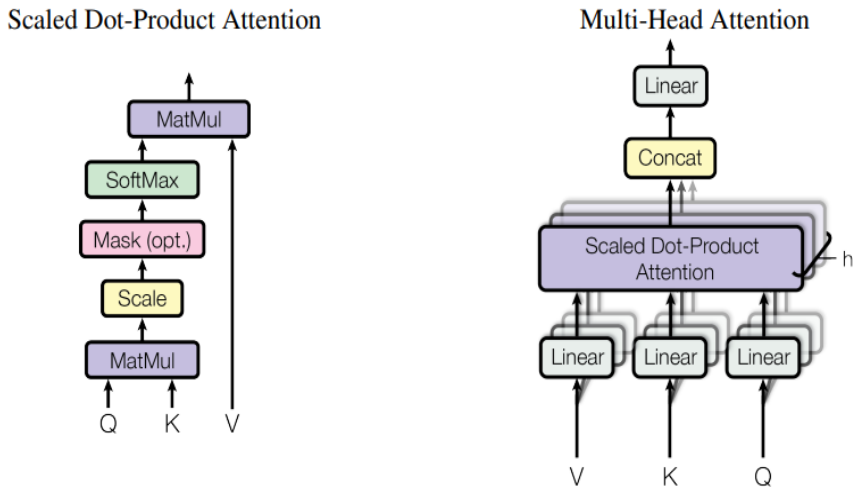


Figure 6.6: *Scaled Dot-Product and Multi-Head Attention Architectures [1]*

Scaled Dot-Product Attention: In the Scaled Dot-Product Attention mechanism, the input comprises queries and keys of dimension d_k, and values of dimension d_v. The dot products of the queries with all keys are calculated, each divided by the square root of d_k, and then a softmax function is applied to obtain weights on the values. This can be represented mathematically as:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{6.10}$$

The two primary attention functions commonly used are additive attention and dot-product (multiplicative) attention. Dot-product attention is notably faster and more space-efficient in practical implementations, owing to its compatibility with highly optimized matrix multiplication code.

Multi-Head Attention: Rather than executing a single attention function with d_{model} − *dimensional* keys, values, and queries, it has been found advantageous to linearly project queries, keys, and values h times using different learned linear projections to d_k, d_k, $and d_v$ dimensions, respectively. For each of these projected versions, the attention function is applied in parallel, producing $d_v - dimensional$ output values. These outputs are then concatenated and subjected to another linear projection to obtain the final values. Multi-head attention allows the model to collectively attend to information from various representation subspaces at different positions. Mathematically, it can be expressed as:

$$MultiHead(Q, K, V) = Concat(head_1, head_2, ..., head_H)W_o \tag{6.11}$$

Where each $head_i$ is calculated as:

$$head_i = Attention(QW_{q,i}, KW_{k,i}, VW_{v,i}) \tag{6.12}$$

In this context, the work employs $h = 8$ parallel attention layers or heads, with each $d_k = d_v = \frac{d_{model}}{h} = 64$. This choice ensures that despite the reduced dimensionality of each head, the overall computational cost remains comparable to that of single-head attention with full dimensionality.

6.6.2 ViT (Vision Transformer)

ViT stands out as the pioneering transformer model specifically tailored for computer vision tasks [9]. It demonstrated the capability of a pure transformer when directly applied to sequences of image patches, showcasing impressive performance in image classification tasks. Through pre-training on extensive datasets and subsequent transfer to various mid-sized or small image recognition benchmarks, Vision Transformer (ViT) achieves remarkable results compared to state-of-the-art convolutional networks. ViT accomplishes this with significantly reduced computational resources. Inspired by the successful scaling of transformers in Natural Language Processing (NLP), the creators of ViT explored the application of a standard transformer directly to images with minimal modifications. This involved dividing an image into patches and presenting the sequence of linear embeddings of these patches as input to a transformer. Image patches are treated like tokens (words) in an NLP context. The model is then trained for image classification in a supervised fashion.

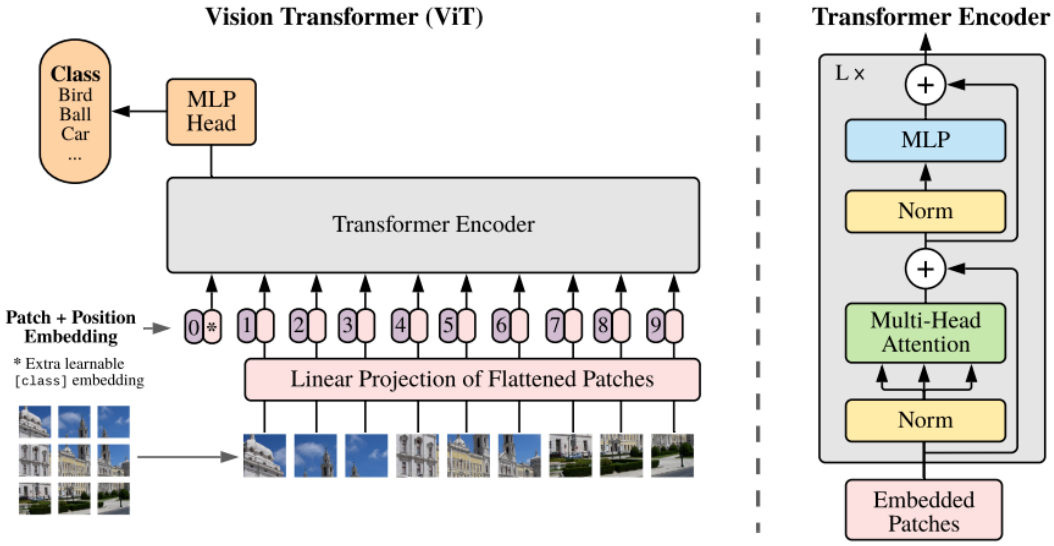


Figure 6.7: *ViT Architecture [9]*

The ViT architecture divides an image into patches of fixed size, linearly embeds each patch, adds position embeddings, and feeds the resulting sequence of vectors into a standard transformer encoder. To facilitate classification, an additional learnable "classi-

fication token" is introduced to the sequence. More specifically, the standard transformer processes a 1D sequence of token embeddings. To accommodate 2D images, the image ($x \in \mathbb{R}^{H\times W\times C}$) is reshaped into a sequence of flattened 2D patches ($x_p \in \mathbb{R}^{N\times(P^2C)}$), where (H, W) represents the original image resolution, C is the channel count, (P, P) is the resolution of each image patch, and $N = \frac{HW}{P^2}$ denotes the resulting number of patches. This also serves as the effective input sequence length for the transformer. Maintaining a constant latent vector size D throughout all layers, the patches are flattened and mapped to D dimensions using a trainable linear projection. The output of this projection is referred to as patch embeddings. Similar to BERT's class token, a learnable embedding is prepended to the sequence of embedded patches, and its state at the output of the transformer encoder serves as the image representation (y). A classification head, implemented by an MLP with one hidden layer during pre-training and a single linear layer during fine-tuning, is attached both in pre-training and fine-tuning. Position embeddings are added to the patch embeddings to retain positional information. Standard learnable 1D position embeddings are used, as no significant performance gains were observed with more advanced 2D-aware position embeddings. The resulting sequence of embedding vectors serves as the input to the encoder, consisting of alternating layers of multiheaded self-attention and MLP blocks. Layernorm (LN) is applied before each block, with residual connections after each block. ViT is typically pre-trained on large datasets and fine-tuned for downstream tasks, involving the removal of the pre-trained prediction head and attachment of a zero-initialized $D\times K$ feedforward layer, where K is the number of downstream classes. Fine-tuning at higher resolution than pre-training is often beneficial. When feeding higher-resolution images, the patch size remains constant, resulting in a larger effective sequence length. While ViT can handle arbitrary sequence lengths (up to memory constraints), adjustments are made for pre-trained position embeddings through 2D interpolation, respecting their location in the original image. It's essential to note that the resolution adjustment and patch extraction are the sole instances where an inductive bias regarding the 2D structure of images is manually introduced into the Vision Transformer.

6.7 Transformers for Segmentation (SOTA)

6.7.1 Swin Transformer

The Swin Transformer [10], designed as a versatile backbone for computer vision, extends the transformer architecture originally developed for Natural Language Processing (NLP) tasks. While transformers excel at modeling long dependencies in input sequences, adapting them from language to vision introduces challenges due to fundamental differences between the two domains. Variances in the scale of visual entities and the high pixel resolution in images compared to words in text necessitate a tailored approach. Addressing these challenges, the Swin Transformer emerges as a hierarchical model that computes representations using shifted windows.

Swin Transformer constructs hierarchical feature maps by initiating from small image

patches, similar to tokens. In deeper layers, it merges neighboring patches, exhibiting linear computational complexity concerning image size. This linear complexity is achieved through self-attention computation in non-overlapping windows, dividing the image, with a predefined number of patches in each window. Shifted windows establish connections among preceding layer windows, significantly enhancing modeling capabilities. Notably, all query patches within a window share the same key set, facilitating memory access in hardware.

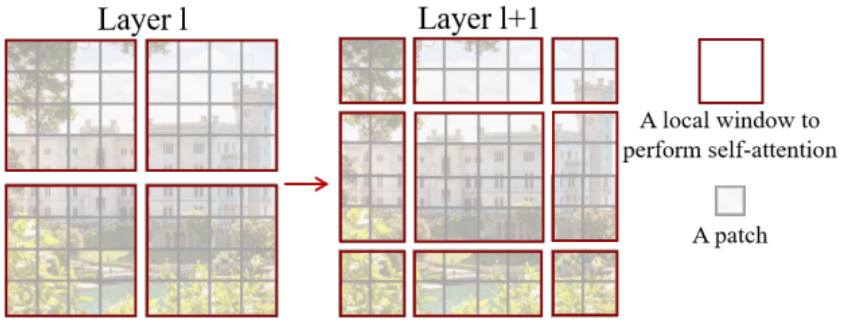


Figure 6.8: *Shifted window approach for computing self-attention in Swin Transformer architecture [10]*

In each layer (l), a regular partitioning scheme performs self-attention computation within each window. Moving to the next layer (l+1), the window partitioning shifts, creating new windows. The self-attention computation in these new windows extends beyond the boundaries of the previous layer's windows, establishing connections among them.

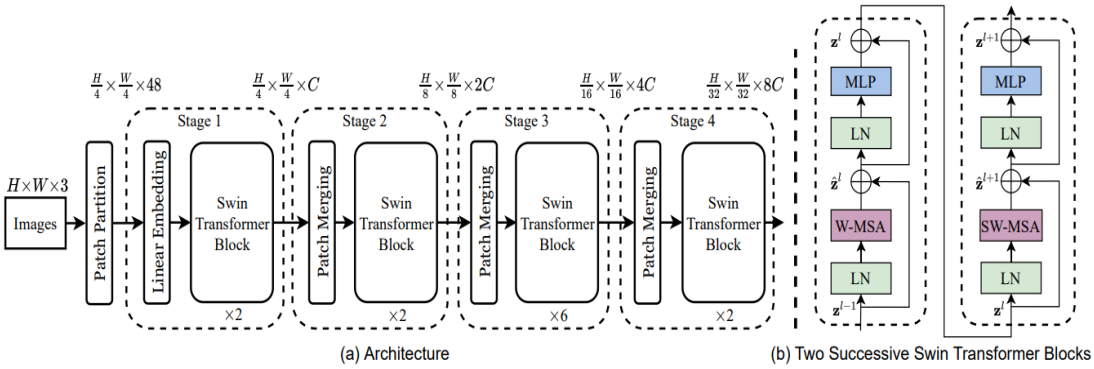


Figure 6.9: *The architecture of a Swin Transformer [10]*

Every patch in the image functions as a token, with its feature derived from the RGB values of its pixels. Using a patch size of 4×4, the feature dimension for each patch is $4 \times 4 \times 3 = 48$. A linear embedding layer is applied to project this feature to an arbitrary dimension, denoted as C. The Swin Transformer applies multiple transformer blocks to these patch tokens, along with the linear embedding, forming "Stage 1." To achieve a hierarchical representation, the number of tokens reduces through patch merging layers as the network deepens.

The initial patch merging layer concatenates features from each group of 2×2 neighboring patches, applying a linear layer to the 4C-dimensional concatenated features. This

results in a reduction of tokens by a factor of $2 \times 2 = 4$ (a $2\times$ downsampling of resolution), with the output dimension set to 2C. Subsequent Swin Transformer blocks transform features, maintaining the resolution at $\frac{H}{8} \times \frac{W}{8}$. This initial block of patch merging and feature transformation constitutes "Stage 2." The process repeats twice, forming "Stage 3" and "Stage 4," with output resolutions of $\frac{H}{16} \times \frac{W}{16}$ and $\frac{H}{32} \times \frac{W}{32}$, respectively.

A Swin Transformer block comprises a shifted window-based MultiHead Self-Attention (MSA) module, followed by a 2-layer Multi-Layer Perceptron (MLP) with GELU nonlinearity in between. LayerNorm (LN) layers are applied before each MSA module and each MLP, with a residual connection applied after each module.

6.7.2 SegFormer

SegFormer [2] represents a straightforward yet efficient semantic segmentation framework that seamlessly integrates Transformers with lightweight Multilayer Perceptron (MLP) decoders. This framework introduces a hierarchically structured Transformer encoder that eliminates the need for additional positional encoding, yielding multiscale features. Departing from intricate transformer decoders, SegFormer opts for an MLP decoder, which strategically aggregates information from diverse layers, incorporating both local and global attention. This approach results in powerful representations, and the simplicity and lightweight design of SegFormer prove instrumental in achieving efficient segmentation.

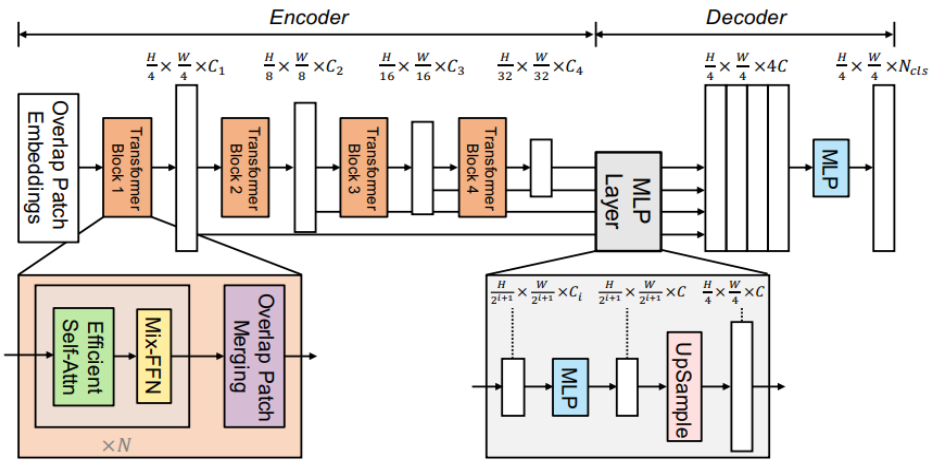


Figure 6.10: *The proposed SegFormer framework [2]*

As illustrated in the figure above, SegFormer comprises two primary modules: Hierarchical Transformer Encoder: This module generates high-resolution coarse features and low-resolution fine features. Contrary to the approach of ViT, which employs patches of size 16×16, SegFormer divides an input image of size $H \times W \times 3$ into smaller patches of size 4×4. The use of these smaller patches is particularly beneficial for dense prediction tasks. The hierarchical Transformer encoder processes these patches, producing multi-level features at $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ of the original image resolution. Lightweight All-MLP Decoder: This module fuses the multi-level features obtained from the encoder to generate the fi-

nal semantic segmentation mask. The image patches serve as input to the hierarchical Transformer encoder, resulting in multi-level features that are then passed to the All-MLP decoder. The decoder predicts the segmentation mask at an $\frac{H}{4} \times \frac{W}{4} \times N_{cls}$ resolution, where N_{cls} represents the number of categories. The combination of a hierarchically structured Transformer encoder and a lightweight All-MLP decoder distinguishes SegFormer's design, showcasing its efficacy in addressing the challenges of semantic segmentation.

6.7.3 MaskFormer

MaskFormer introduces a straightforward mask classification model designed to predict a set of binary masks, each tied to a singular global class label prediction [3]. Its fundamental insight lies in the versatility of mask classification, offering a unified solution for both semantic and instance-level segmentation tasks through a shared model, loss, and training procedure. Notably, MaskFormer demonstrates superior performance over per-pixel classification baselines, particularly in scenarios with a large number of classes. While many deep learning-based semantic segmentation approaches frame the problem as per-pixel classification, applying a classification loss to individual output pixels, MaskFormer adopts an alternative paradigm. It disentangles the image partitioning and classification aspects of segmentation by predicting a set of binary masks, each associated with a single class prediction. Leveraging the set prediction mechanism from DETR, MaskFormer employs a Transformer decoder to compute pairs, each comprising a class prediction and a mask embedding vector. The mask embedding vector facilitates binary mask prediction through a dot product with the per-pixel embedding obtained from an underlying fully-convolutional network.

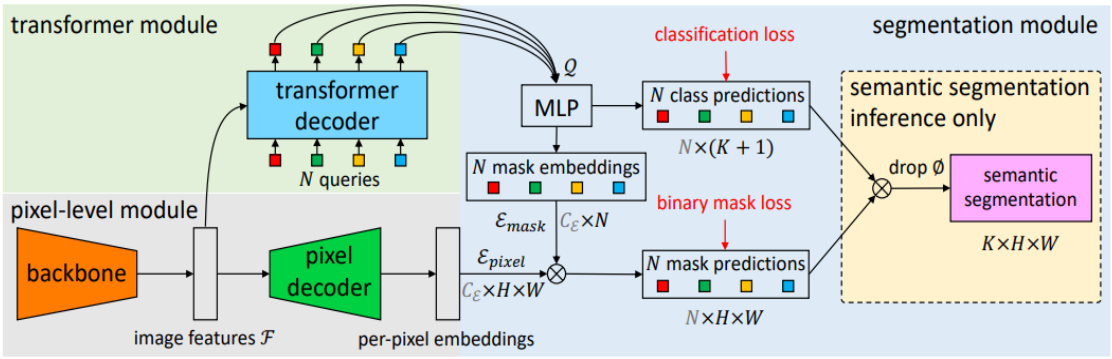


Figure 6.11: *MaskFormer overview architecture [3]*

MaskFormer computes a collection of N probability-mask pairs, denoted as $z = \{(p_i, m_i)\}_{i=1}^{N}$. Illustrated in the figure above. The model consists of three key modules. The Pixel-Level Module extracts per-pixel embeddings, crucial for generating binary mask predictions. The Transformer Module in which a stack of Transformer decoder layers computes N per-segment embeddings, contributing to the subsequent prediction steps. Finally, the Segmentation Module which is responsible for generating predictions $\{(p_i, m_i)\}_{i=1}^{N}$ from the obtained embeddings. This module ensures a cohesive integration of class predictions and binary masks. MaskFormer's architecture showcases the synergy of these modules,

demonstrating its efficacy in addressing both semantic and instance-level segmentation tasks within a unified framework.

6.7.4 Mask2Former

The Masked-attention Mask Transformer (Mask2Former) [4] is a versatile architecture designed to tackle various image segmentation tasks, including panoptic, instance, or semantic segmentation. Key to its performance are masked attention mechanisms that extract localized features by confining cross-attention within predicted mask regions. Mask2Former outperforms specialized architectures in diverse segmentation tasks, ensuring ease of training across different objectives. Comprising a backbone feature extractor, a pixel decoder, and a Transformer decoder, Mask2Former introduces several improvements for enhanced results and efficient training.

The first improvement involves utilizing masked attention in the Transformer decoder, restricting attention to features centered around predicted segments. This localized focus, whether on objects or regions, contrasts with the cross-attention of a standard Transformer decoder, leading to faster convergence and improved performance. Second, Mask2Former leverages multi-scale high-resolution features, enhancing its ability to segment small objects or regions. Third, optimization enhancements, such as reversing the order of self and cross-attention, making query features learnable, and removing dropout, contribute to improved performance without additional computational cost. Lastly, the model achieves a 3× reduction in training memory without compromising performance by calculating mask loss on a few randomly sampled points. These enhancements not only elevate model performance but also significantly simplify training, making universal architectures more accessible to users with limited computational resources.

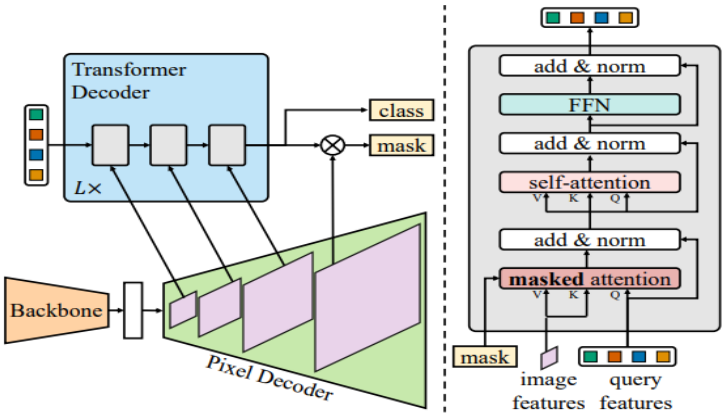


Figure 6.12: *Mask2Former overview architecture [4]*

Mask2Former adopts the same meta-architecture as MaskFormer, featuring a backbone, a pixel decoder, and a Transformer decoder. Notably, the Transformer decoder introduces masked attention instead of the conventional cross-attention. To address small objects effectively, a practical approach involves utilizing high-resolution features from a pixel decoder by feeding one scale of the multi-scale feature to one Transformer

decoder layer at a time.

6.8 Metrics in semantic segmentation

As indicated in [24], semantic segmentation is a challenging and complex task that considers the relationships among classified pixels. Throughout the discussion, any mention of "class" is interchangeable with "category." The pixel-wise accuracy (pACC) serves as an initial metric to assess performance and it is computed with the following equation:

$$acc = \frac{\sum_{i=1}^{k} n_{ij}}{\sum_{i=1}^{k} t_i} \tag{6.13}$$

where n_{ij} is the number of pixels which belong to class i and were labeled as class j, k is the total number of classes and $t_i = \sum_{j=1}^{k} n_{ij}$ is the total number of pixels of class i. Nevertheless, this metric can be misleadingly high in datasets where extensive regions are dominated by a single class. In such instances, the model may have merely learned the prevalent occurrences of certain elements in specific areas of the image. This issue can be addressed by considering the following metrics:

mACC Mean accuracy is the mean accuracy across all classes:

$$mACC = \frac{1}{k} \sum_{i=1}^{k} \frac{n_{ij}}{t_i} \tag{6.14}$$

IoU Intersection over Union measures the overlap between the predicted segmentation and the ground truth segmentation for each class, providing insights into how well the model's predictions align with the actual boundaries of objects or regions in the image. The metric IoU is a per class assessment on the intersection of the inferred segmentation and the ground truth, divided by the union excluding pixels labelled as "void":

$$IoU = \frac{TruePositive_i}{TruePositive_i + FalsePositive_i + FalseNegative_i} \tag{6.15}$$

where $TruePositive_i$, $FalsePositive_i$, and $FalseNegative_i$ represent the numbers of true positive, false positive, and false negative pixels, respectively, for class i.

mIoU (Mean Intersection over Union) It provides a measure of the overall segmentation performance by calculating the average Intersection over Union (IoU) across all classes or categories. mIoU is then calculated as the mean of IoU values across all classes:

$$mIoU = \frac{1}{k} \sum_{i=1}^{k} IoU_i \tag{6.16}$$

where k is the total number of classes or categories.

F1 Score: The F1 score is a measure of a model's accuracy, balancing both precision and recall. It is calculated as the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{6.17}$$

where precision is the ratio of true positive predictions to the total number of positive predictions, and recall is the ratio of true positive predictions to the total number of actual positives.

Chapter 7

Experimental Part

7.1 Datasets

7.1.1 Cityscapes Dataset

The Cityscapes Dataset [19] is specially crafted for autonomous driving applications within urban settings, offering a diverse collection of images showcasing road environments across 50 distinct city centers throughout all four seasons. These images come in both low (8-bit) and high (16-bit) resolutions. The dataset encompasses pixel-level and instance-level annotations, categorized into coarse and fine groups. Fine annotations meticulously label every pixel within polygons of instances across 5000 images spanning 27 cities. Conversely, coarse annotations, applied to images from 23 cities, prioritize speed in polygon selection, resulting in slightly less accurate labeling. With a total of 30 classes, 19 of which are utilized in evaluations, this dataset provides a comprehensive depiction of inner-city roads, considering factors such as traffic and diverse climatic conditions. Its uniqueness in the realm of autonomous vehicle driving lies in its broad representation and wide variety of classes. Notably, in terms of instance-level annotations, Cityscapes stands out as the sole dataset including instances of people and vehicles.

7.1.2 ADE20k

ADE20k [23] was developed in response to the recognition of a need for a comprehensive dataset that encompasses a wide range of scenes and common objects. Prior datasets, such as Cityscapes, were limited in scene diversity, while others like COCO and Pascal focused on only a few or insignificant object classes. ADE20k addresses this gap by offering a dataset comprising 20,210 images in the training set, 2,000 images in the validation set, and 3,000 images in the testing set. Each image is meticulously annotated with objects, and many objects also include annotations for their respective parts. Moreover, the annotations provide additional information, such as whether an object is cropped, along with other attributes. Notably, parts can themselves have subparts, and these associations are appropriately labeled. This comprehensive approach in ADE20k ensures a more detailed and nuanced understanding of scenes and objects, distinguishing it from earlier datasets with more limited scope and coverage.

7.1.3 Coast Train Dataset

Initial Data Description

"Coast Train" [25] is a multi-label dataset of orthomosaic and satellite images capturing diverse coastal environments, complemented by corresponding labels. Each dataset within "Coast Train" is specifically associated with a unique image type and class set. In terms of spatial resolution, the class sets exhibit a horizontal range from 0.05m to 1m for orthomosaics, while satellite imagery encompasses resolutions of either 10m or 15m. Notably, the inclusion of orthomosaic imagery provides a fine-grained representation of specific coastal environments, boasting an impressive 5-cm pixel resolution. The dataset features a variety of imagery sources, including NAIP (1m), Quadrangle (6m), Sentinel-2 (10m), and Landsat-8 (15m), collectively offering a continental-scale perspective on the diversity of coastal environments. This diverse array of pixel resolutions ensures a comprehensive representation of coastal features at different scales.

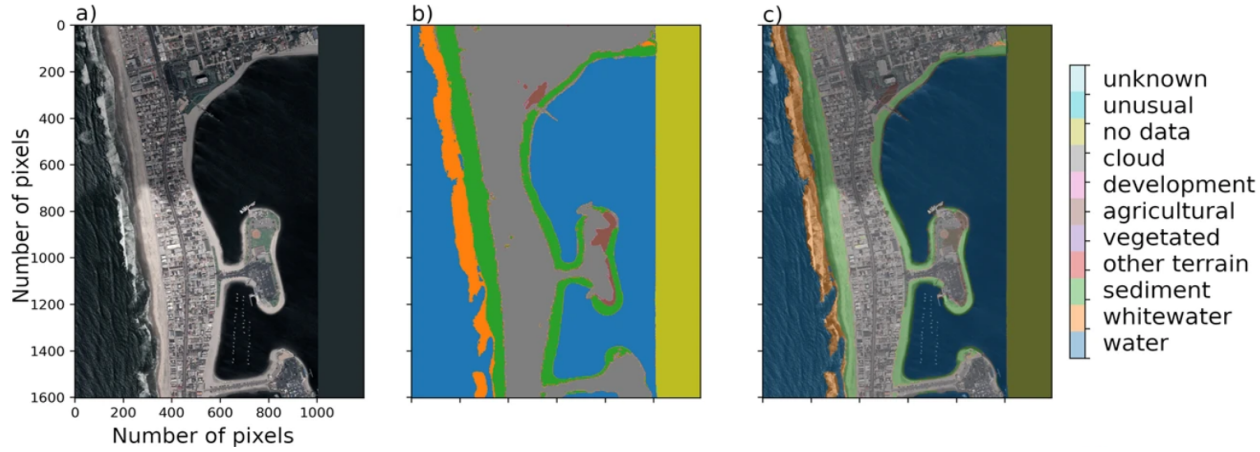


Figure 7.1: *This figure depicts one example image (a), corresponding label image (b), and image-label overlay (c), of one of the orthomosaic datasets. This particular example shows imagery from San Diego, California*

Class labels within the dataset range between 4 and 12, contributing to a detailed annotation of the images. In total, the dataset comprises 1852 individual images, comprising 1.196 billion pixels. This vast dataset represents a total surface area of 3.63 million hectares, providing a rich resource for studying and understanding coastal ecosystems. The dataset's geographical scope is extensive, spanning from 26 to 48 degrees N in latitude and 69 to 123 degrees W in longitude, as illustrated in Figure 7.2. This broad coverage ensures that the dataset captures a wide spectrum of coastal environments, making it a valuable resource for research and analysis in the field.

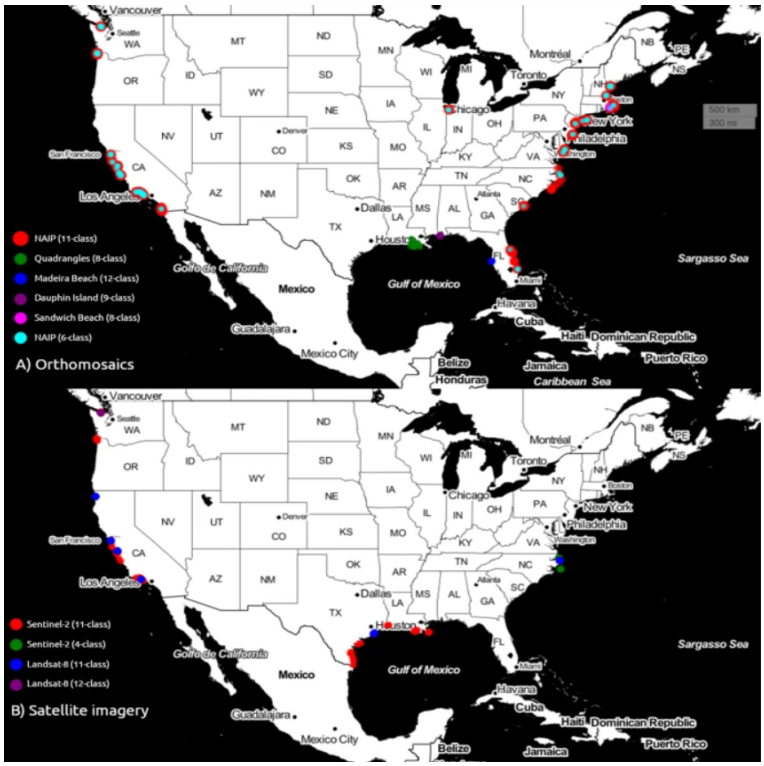


Figure 7.2: *Geographical distribution of (A) orthomosaic and (B) satellite imagery*

Each data record has a unique set of classes; however, labels are easily re-processed to map multiple classes to a standardized set of "superclasses" across all data records. Superclasses, as shown in Table 7.1, are broad class names for a collection of component class labels. For example, 'buildings' and 'vehicles' are a subset of the 'developed' superclass, and 'sand' and 'gravel' are part of the 'sediment' superclass. Seven superclass labels, and between four and twelve class labels depending on the dataset, are defined.

Superclass Mapping	
Superclass Names	**Aliases (component class names)**
water	water, sediment plume
whitewater	whitewater, surf
sediment	sediment, sand, gravel, gravel/shell, cobble/boulder/ mud/silt
developed	developed, dev, coastal defense, pavement/road, other anthro, vehicles, buildings, development
natural terrain	bedrock, bare ground, other natural terrain, other bare natural terrain
vegetation	vegetated, vegetated surface, vegetated ground, terrestrial vegetation, marsh vegetation, herbaceous veg, herbaceous vegetation, wood vegetation, woody veg
other	other, unknown, unusual, nodata, people, ice/snow, cloud

Table 7.1: *Mapping from per-set classes to superclasses.*

Dataset Preparation

The coastTrain dataset comprises ten distinct sub-datasets, namely Landsat8-11-001, Landsat8-12-001, NAIP-11-001, NAIP-6-001, Orthophoto-12-001, Orthophoto-8-001, Orthophoto-9-001, Quadrangles-7-001, Sentinel2-11-001, and Sentinel2-4-001. These sub-datasets present a diverse array of coastal environment images, varying in resolution, size, and content. In order to create a dataset tailored to the specific requirements of our semantic segmentation task, a detailed inspection of each sub-dataset was conducted. In this process, we excluded images with dimensions below 300x300 since their smaller size could compromise the model's ability to capture crucial details for our segmentation task. Additionally, images deemed irrelevant or non-contributory to the objectives of this study were excluded from the selection. For instance, the NAIP and Quadrangles sub-datasets were excluded because they would have a negative effect on the training process, increasing the training time of the models and not contributing to their results. With the above process, out of the initial pool of 1852 images, we kept 645 images. All selected images were resized to a uniform dimension of 512x512 pixels, for training purposes. The above preprocessing pipeline ensures that the coastTrain dataset employed in our study is optimized for training transformer models on semantic segmentation tasks related to coastal environments, striking a balance between inclusivity and precision.

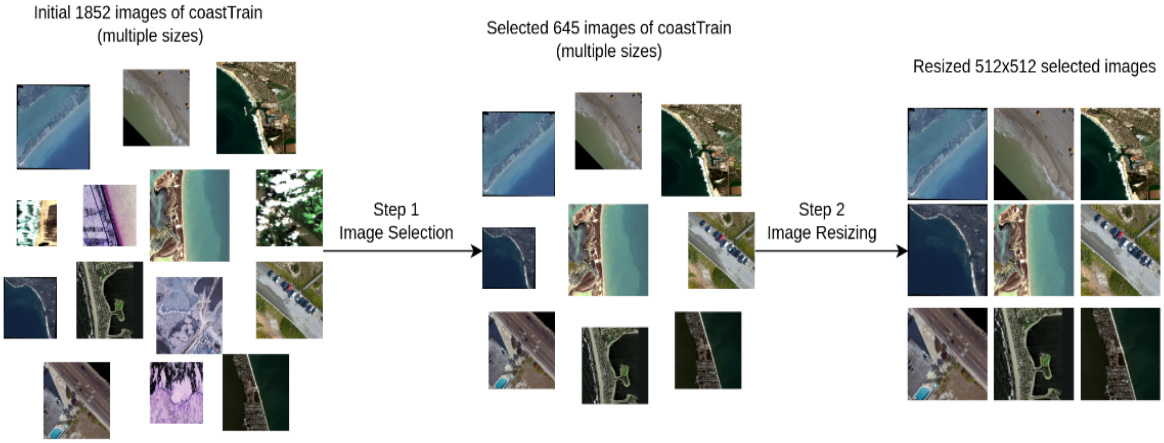


Figure 7.3: *Preprocessing pipeline of coastTrain dataset. Step 1: We select the best images according to size, resolution and compatibility with our task. Step 2: We resize all the images to size 512x512.*

The final coastTrain dataset consists of 645 images, each with a resolution of 512x512 pixels, accompanied by their respective masks. To ensure consistency in the dataset for effective training, we have mapped all classes to their corresponding superclass (Table 7.1). This mapping helps create a uniform dataset tailored to meet the specific requirements of our semantic segmentation task. The dataset encompasses seven distinct classes, as outlined in the table below (Table 7.2). Each pixel in the images can be assigned to one of these classes, contributing to the richness of information captured for accurate semantic segmentation.

Classes	
Id	**Label**
0	water
1	whitewater
2	sediment
3	development
4	natural terrain
5	vegetation
6	unknown

Table 7.2: *Id to label table for the final Coast Train dataset.*

7.1.4 Greek Coastline Dataset (From Greek Land Registry)

Figure 7.4: *Example image from Greek Coastline Dataset*

Initial Data Description

The Greek Coastline dataset consists of aerial images at 25cm pixel resolution and it was provided to us by the Greek Land Registry. Data files are in raster format, thus we used the python Rasterio library [26] to read them and process them. More specifically, each image consists of four separate files (.tif, .tfw, .aux, .img), the combination of which forms the final three-dimensional image (longitude, latitude, altitude). The .tif, .tfw and .aux files contain elements that compose the two-dimensional image, while the .img file contains information about the third dimension of the image, namely, the height. In particular, the .tif file includes the spectrum of the two-dimensional image in four separate spectral bands (Red, Green, Blue, Infrared). The .tfw file contains the transformation data that needs to be applied to each pixel of the image to derive the corresponding geographic longitude and latitude. The .aux file includes all metadata related to the two-dimensional image, such as the coordinate reference system used by the image to express the geographic longitude and latitude of its elements. Finally, the .img file contains the third dimension of the image, i.e., the data for the height of each pixel in the two-dimensional image, while also including the corresponding metadata related to this dimension. The

height and width of the image may vary slightly from image to image. We choose to crop the extra pixels that may exist in some images to have a uniform dataset consisting of images with dimensions of 3200x2400. The pixel resolution of the .tif files (images) is 25cm, while for the .img files (height) is 20m. So, we have a 1/4 ratio between image and height files. Therefore, to achieve alignment in the coordinates of the two images, we reshape the height matrix to have dimensions 3200x2400 instead of 800x600. We achieve this by using the Kronecker product, multiplying each element of the height matrix by a 4x4 neighborhood matrix.

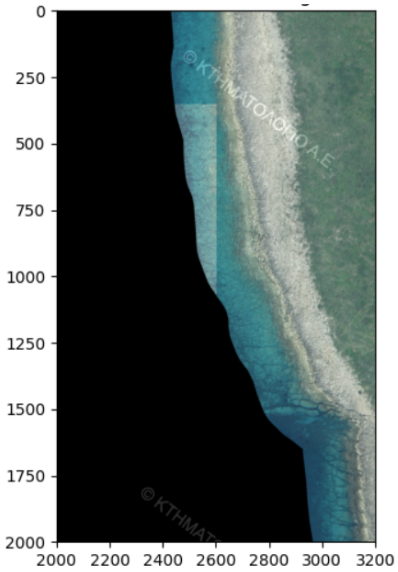


Figure 7.5: *2D example image*

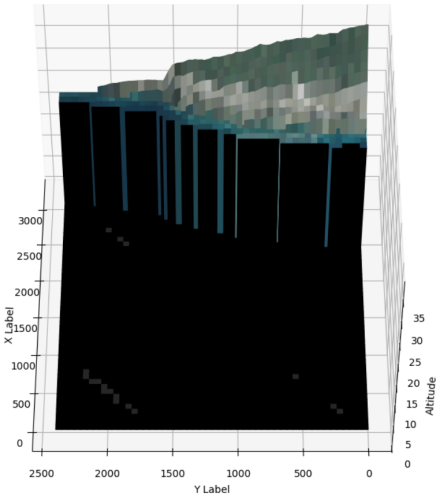


Figure 7.6: *3D example image*

From the .tfw file we acquire the coordinate transformation that transforms the pixel coordinates to actual geographic coordinates. More specifically the structure of the transformation is shown in equation 7.1 below.

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} A & B \\ D & -E \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} C \\ F \end{bmatrix} \tag{7.1}$$

Where:

- x_1 is the calculated x-coordinate of the pixel on the map.
- y_1 is the calculated y-coordinate of the pixel on the map.
- x is the column number of a pixel in the image.
- y is the row number of a pixel in the image.
- A is the x-scale, the dimension of a pixel in map units in the x-direction.
- B and D are the rotation terms.
- C and F are the translation terms, representing the x, y map coordinates of the center of the upper left pixel.

- *E* is the negative of y-scale, the dimension of a pixel in map units in the y-direction.

For our dataset the values of fixed parameters A, B, D, E are shown in table 7.3 below. Parameters C and F are different for every image as they correspond to the longitude and latitude of the upper left pixel.

Transformation Parameters	
Name	**Value**
A	0.25
B	0.00
D	0.00
E	-0.25

Table 7.3: *Transformation parameters to convert pixel coordinates to HGRS87.*

Using this transformation, the coordinates of pixels in the image are mapped to the corresponding geographic longitude and latitude. The resulting geographic coordinates are expressed in the Hellenic Geodetic Reference System 1987 (HGRS87), as indicated by the metadata in the .aux file.

Dataset Preparation

To ensure the suitability of images for training in accordance with our model's input restrictions, we implement a meticulous preprocessing strategy. All models utilized in our study employ a consistent input size of 512x512 pixels. However, for the Greek Coastline dataset, we adopt a distinctive approach to preserve the maximum possible resolution, given the intricate nature of the coastal segmentation task. In order to achieve this, we opt for image splitting rather than resizing in order to maintain the highest resolution as it is crucial for the demands of coastal segmentation. The specific preprocessing pipeline we follow is visually depicted in Figure 7.7, and it involves two key steps. Firstly, we initiate the preprocessing pipeline by zero-padding the original images, which are initially sized at 3200x2400 pixels. This zero-padding is applied both horizontally and vertically to ensure that the dimensions are precisely divisible by 512. So after this step the size is set to 3584x2560 for every image. Secondly, the zero-padded image is splitted into 35 512x512 images.

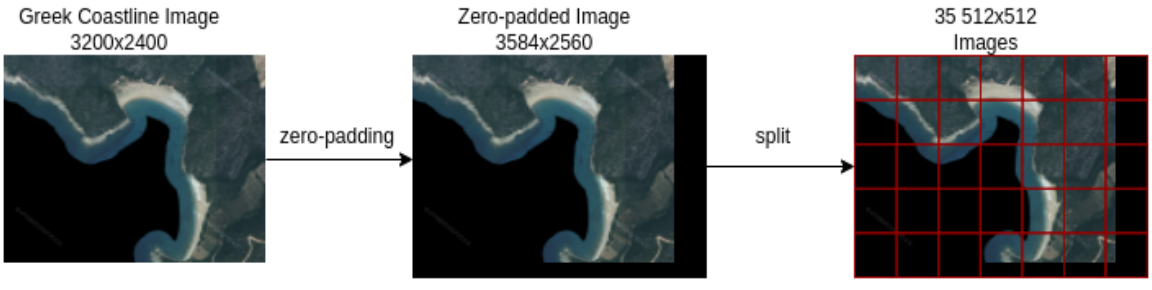


Figure 7.7: *Image preprocessing: Zero-padding and splitting to patches*

Figure 7.8: *Example patch*

The objective of this project is to segment the Greek Coastline dataset, which was initially unlabelled and comprised raw images. To enhance the quality of our results, we chose to manually label a select subset of the dataset using the Segments.ai tool—an effective training data platform for computer vision engineers, offering a powerful interface for data labeling. We decided to label 420 512x512 images (which corresponds to 12 full images) utilizing the same set of labels as the coastTrain dataset, which is illustrated in Table 7.2 above.

Figure 7.9: *Example original image with corresponding mask (black: unknown, blue: water, white: whitewater, yellow: sediment, green: vegetation)*

7.2 Architecture

7.2.1 Introduction

The architectural framework employed in this thesis aims to achieve semantic segmentation in coastal images of the Greek Coastline dataset described in 7.1.4. Semantic segmentation in coastal images holds significant importance for various applications such as environmental monitoring, coastal management, and disaster response. The objective was to utilize state-of-the-art transformer models, namely SegFormer, MaskFormer, and Mask2Former, to classify each pixel in the images into one of seven distinct classes: water, whitewater, sediment, vegetation, development, other natural terrain, and unknown. However, the significant challenge was that the Greek Coastline dataset was entirely unlabeled, rendering it unsuitable for direct training. To address this challenge, three different approaches were pursued, each aimed at adapting the models to the characteristics of the Greek Coastline dataset.

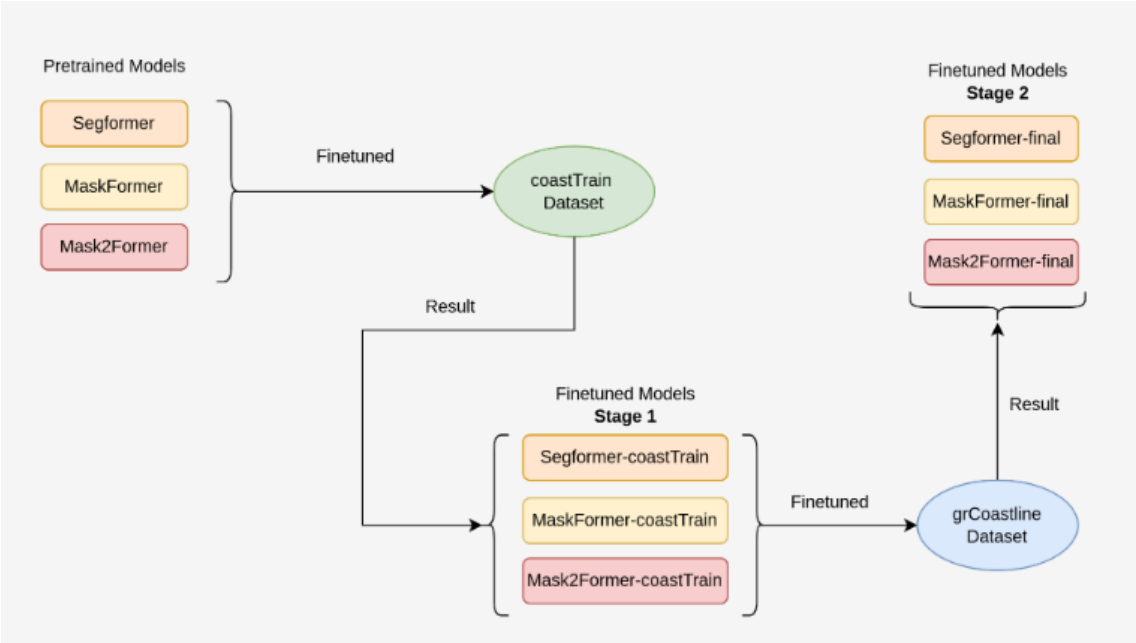


Figure 7.10: *Overall Architecture*

7.2.2 Stage 1 Training

The first approach involved searching for a pre-labeled dataset of coastal environments sharing similar features with our dataset and containing labels relevant to our segmentation task. In this pursuit, the coastTrain dataset [25] was discovered. The dataset underwent preprocessing to align with the requirements for training, a process thoroughly described in Chapter 7.1.3. Regarding the transformer models, pretrained models tailored for semantic segmentation tasks were sought to enhance the performance of our models. Given the resource-intensive nature of transformer models, pretrained models trained on expansive datasets like Cityscapes [19] and ADE20k [23] were identified. These models were subsequently fine-tuned using the coastTrain dataset, resulting in the development of stage 1 models. Stage 1 models will be employed for inference on the Greek Coastline dataset to visually evaluate the segmentation results, which are displayed in Chapter **??**. These models will also serve as backbones for further training in the next stage.

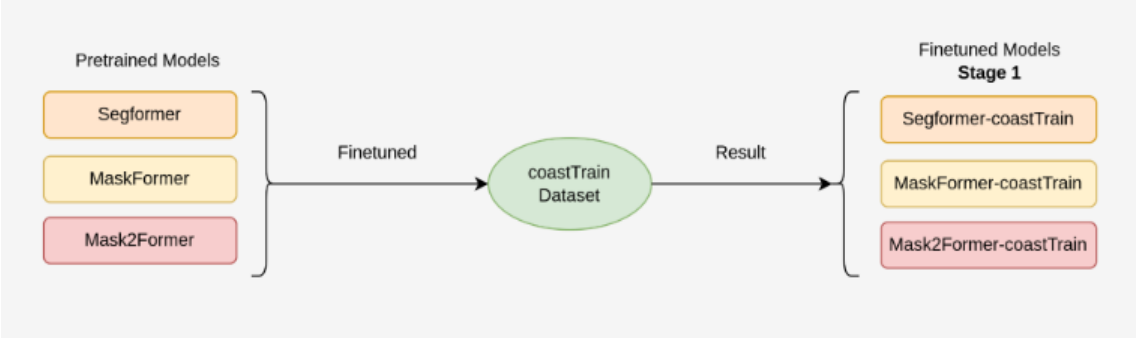


Figure 7.11: *Stage 1 Architecture*

7.2.3 Stage 2 Training

Upon evaluating the stage 1 models on the Greek Coastline dataset, it became evident that further fine-tuning was necessary to adapt the models to the specific characteristics of the dataset. In the second approach, the results of stage 1 models were improved by further fine-tuning them to a subset of the Greek dataset that we manually labeled, as explained in Chapter 7.1.4. This subset contains the same labels as the coastTrain dataset. Thus, the stage 1 models fine-tuned on the coastTrain dataset were further fine-tuned to the Greek Coastline dataset to adjust the models' predictions to our own images. This training process led to the creation of Stage 2 models, which we also use for inference to visually examine their results compared to stage 1 models.

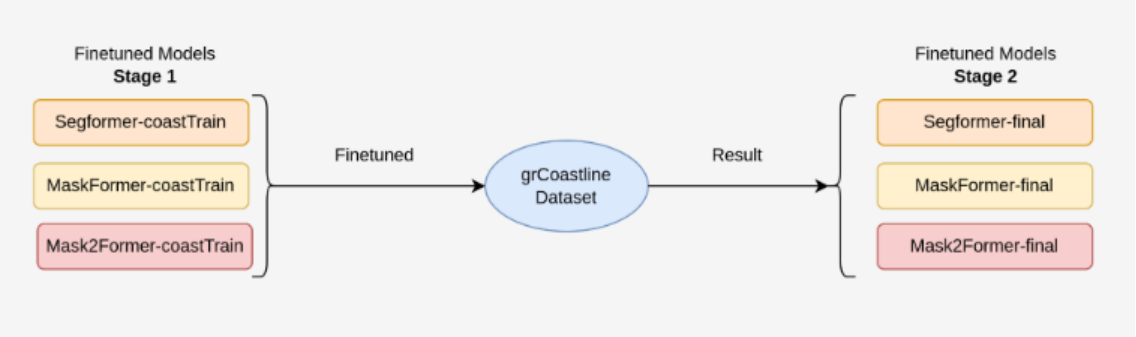


Figure 7.12: *Stage 2 Architecture*

7.2.4 Direct Greek Coastline Training

In the third approach, pretrained models were finetuned directly on the Greek Coastline dataset for comparison reasons to evaluate the significance of using the coastTrain dataset in the previous approach. This approach aimed to determine whether the step of searching for a pre-labeled coastal dataset to first train the models played a significant role in our results.

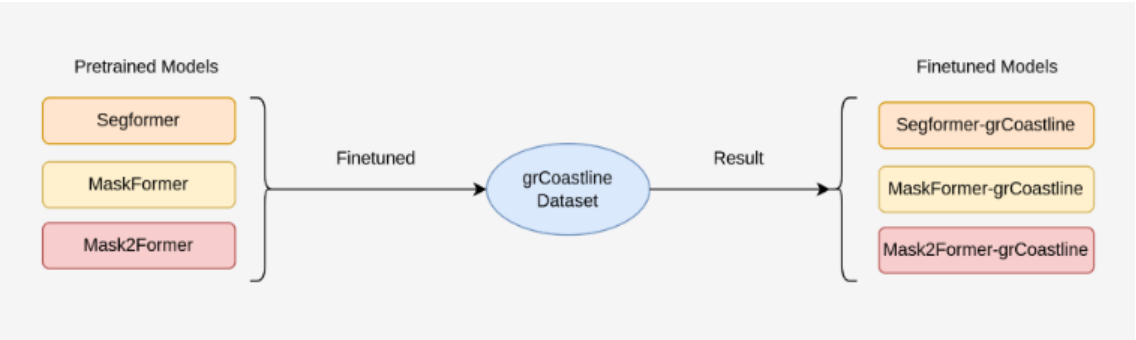


Figure 7.13: *Direct Architecture*

7.3 Results

In the following tables we present the results of the training process for each stage and for all three models. More specifically, we present the values of all the metrics

described in chapter 6.8, but we use the mIoU as the evaluation metric which is the most suitable one for the task of semantic segmentation. We examine different versions for each model concerning their size and dataset used for pretraining. For each training stage we compare the different versions of the models and also the best version of each model with each other.

7.3.1 Stage 1 - coastTrain dataset

SegFormer

For the SegFormer we trained in total 4 different versions of the model. As mentioned in [2], there is a scaling approach for SegFormer which has a series of models from SegFormer-B0 to SegFormer-B5, reaching significantly better performance and efficiency than previous counterparts. For example SegFormer-B4 reaches 51.1% mIoU on ADE20K while SegFormer-B5 reaches 51.8%. On Cityscapes, SegFormer-B4 reaches 83.8% and SegFormer-B5 84.0%. We chose to finetune SegFormer-B4 and SegFormer-B5 pretrained on both Cityscapes and ADE20K in order to evaluate their performance.

Stage 1 SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	79.20	86.09	93.54
SegFormer-B4	ADE20K	76.15	85.69	92.83
SegFormer-B5	Cityscapes	**82.69**	90.60	94.23
SegFormer-B5	ADE20K	77.37	85.45	92.45

Table 7.4: *Stage 1 results of SegFormer models*

As expected, SegFormer-B5 performed better than SegFormer-B4. Furthermore, the models which were pretrained on Cityscapes achieved better results than the ones pretrained on ADE20K, which is also justifiable as ADE20K is a more complex dataset compared to Cityscapes which suits better the coastTrain data. The best SegFormer model on Stage 1 is the SegFormer-B5 pretrained on Cityscapes and achieves 82.69% mIoU on the coastTrain dataset.

MaskFormer

For the MaskFormer we trained 2 versions of the model concerning the size of the backbone that they use, both pretrained on ADE20K. The MaskFormer-base utilizes a Swin-B encoder while the MaskFormer-large a Swin-L one. More specifically, we finetuned on coastTrain the MaskFormer-base and MaskFormer-large models, both pretrained on ADE20K. According to [3], the MaskFormer-Base achieves 53.9% mIoU on ADE20K, while the MaskFormer-Large 55.6% mIoU.

Stage 1 MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE20K	81.4	89.64	93.96
MaskFormer-Large	ADE20K	**82.18**	91.76	94.79

Table 7.5: *Stage 1 results of MaskFormer models*

As expected, MaskFormer-Large model pretrained on ADE20K performed better MaskFormer-Base, achieving 82.18% mIoU on coastTrain.

Mask2Former

For the Mask2Former model we finetuned the one with the Swin-L backbone (namely Mask2Former-Large). The first version is pretrained on Cityscpaes while the second one on ADE20K. According to [4], on ADE20K Mask2Former-Large achieves 57.7% mIoU while on Cityscapes it achieves 83.3% mIoU.

Stage 1 Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**84.06**	93.42	95.45
Mask2Former-Large	ADE20K	83.92	91.14	94.67

Table 7.6: *Stage 1 results of Mask2Former models*

The Mask2Former pretrained on Cityscapes performed better than the one pretrained on ADE20K. This outcome is justifiable for two main reasons. Firstly, the mIoU scores of the pretrained models differ substantially, as the Mask2Former achieves 83.3% mIoU on Cityscapes and 57.7% mIoU on ADE20K, mainly due to the complexity and larger number of classes that ADE20K has. Secondly, the Cityscapes dataset is way more similar to coastTrain dataset than ADE20K, as it shares some classes and also the type of content is more compatible with our task.

7.3.2 Direct Training - grCoastline dataset

Continuing with the direct training with the grCoastline dataset, we utilized the same models as in stage 1. The results followed the same pattern as the same models performed better. The training scores for each model are presented in the following tables.

SegFormer

Direct Training SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	72.21	77.96	94.29
SegFormer-B4	ADE20K	72.07	78.64	94.37
SegFormer-B5	Cityscapes	**72.46**	78.87	94.56
SegFormer-B5	ADE20K	69.51	77.38	93.81

Table 7.7: *Direct training results of SegFormer models*

SegFormer-B5 pretrained on Cityscapes achieved the highest score with 72.46% mIoU.

MaskFormer

Direct Training MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE20K	75.55	80.24	93.46
MaskFormer-Large	ADE20K	**78.79**	84.48	94.27

Table 7.8: *Direct training results of MaskFormer models*

MaskFormer-Large pretrained on ADE20K achieved the highest score with 78.79% mIoU.

Mask2Former

Direct Training Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**82.42**	91.28	96.2
Mask2Former-Large	ADE20K	81.91	89.03	93.22

Table 7.9: *Direct training results of Mask2Former models*

Mask2Former-Large pretrained on Cityscapes achieved the highest score with 82.42% mIoU.

7.3.3 Stage 2 - Both datasets

Finally, in Stage 2 we also used the same versions of the models as in the earlier stages. The results again, as we expected, followed the same principles and are presented in the tables below.

SegFormer

Stage 2 SegFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
SegFormer-B4	Cityscapes	74.77	83.06	92.47
SegFormer-B4	ADE20K	75.78	83.13	94.51
SegFormer-B5	Cityscapes	**76.81**	84.19	94.79
SegFormer-B5	ADE20K	72.61	79.41	94.57

Table 7.10: *Stage 2 results of SegFormer models*

SegFormer-B5 pretrained on Cityscapes achieved the highest score with 76.81% mIoU.

MaskFormer

Stage 2 MaskFormer				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
MaskFormer-Base	ADE20K	79.91	86.27	93.46
MaskFormer-Large	ADE20K	**80.59**	89.93	93.87

Table 7.11: *Stage 2 results of MaskFormer models*

MaskFormer-Large pretrained on ADE20K achieved the highest score with 80.59% mIoU.

Mask2Former

Stage 2 Mask2Former				
Model	**Pretrained**	**mIoU (%)**	**Mean Accuracy (%)**	**f1 score (%)**
Mask2Former-Large	Cityscapes	**85.43**	94.33	96.27
Mask2Former-Large	ADE20K	83.82	92.56	94.97

Table 7.12: *Stage 2 results of Mask2Former models*

Mask2Former-Large pretrained on Cityscapes achieved the highest score with 85.43% mIoU.

7.4 Inference

Finally in order to visually evaluate the model's performance we inference the best version of each model from the training process, on images that do not belong in training dataset.

7.4.1 SegFormer

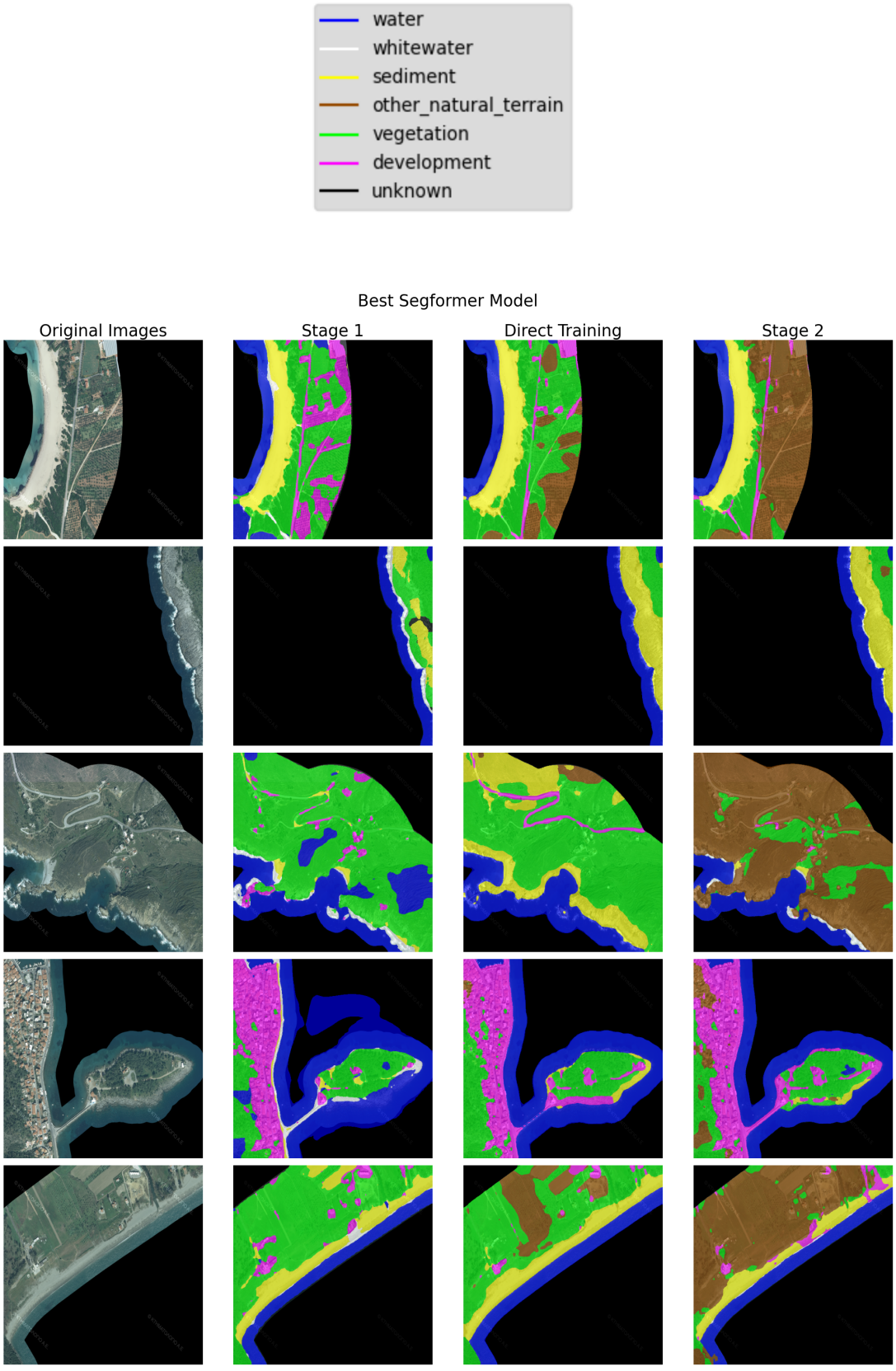


Figure 7.14: *Inference of best SegFormer model on example images of grCoastline*

7.4.2 MaskFormer

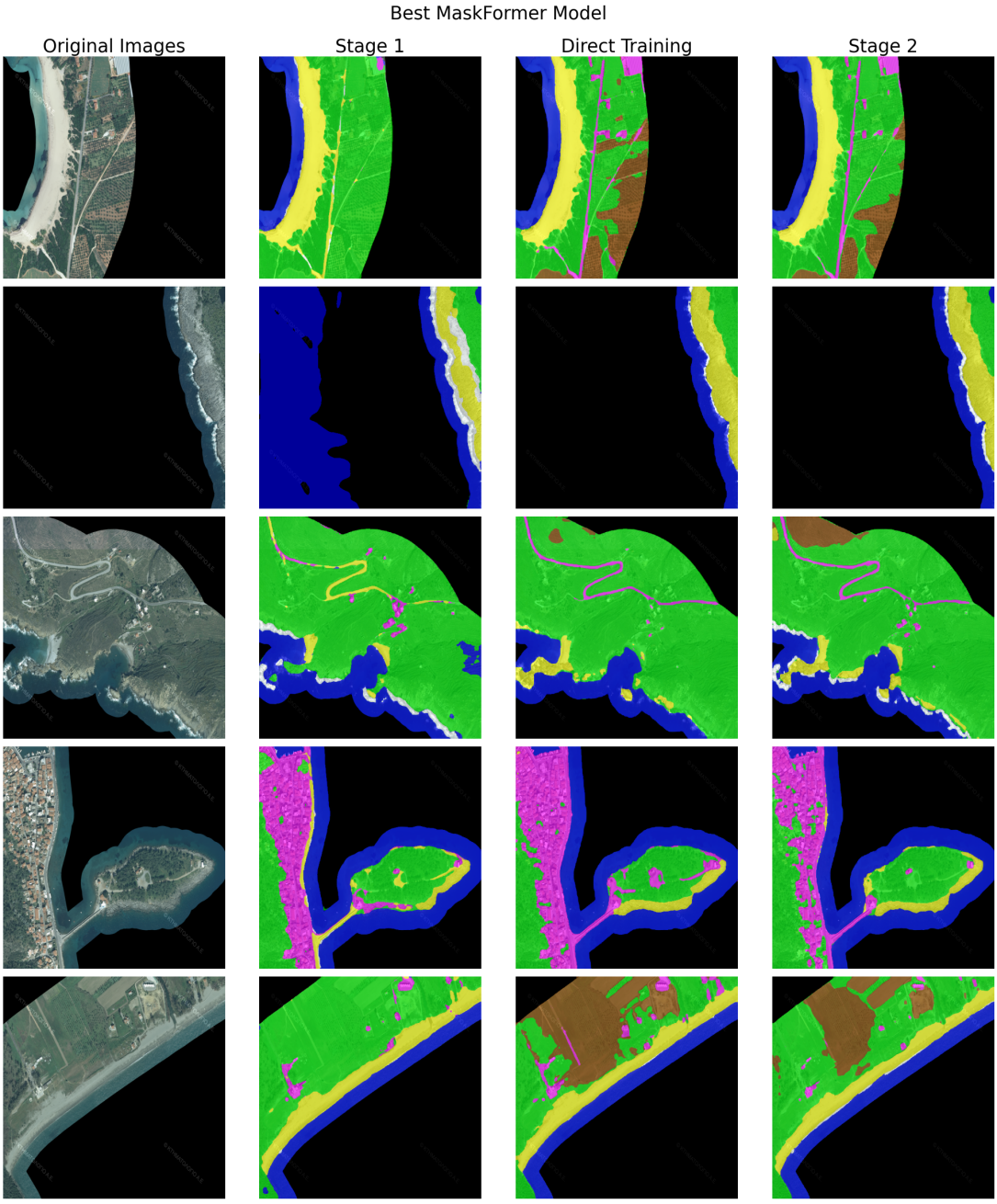


Figure 7.15: *Inference of best MaskFormer model on example images of the Greek Coastline dataset.*

7.4.3 Mask2Former

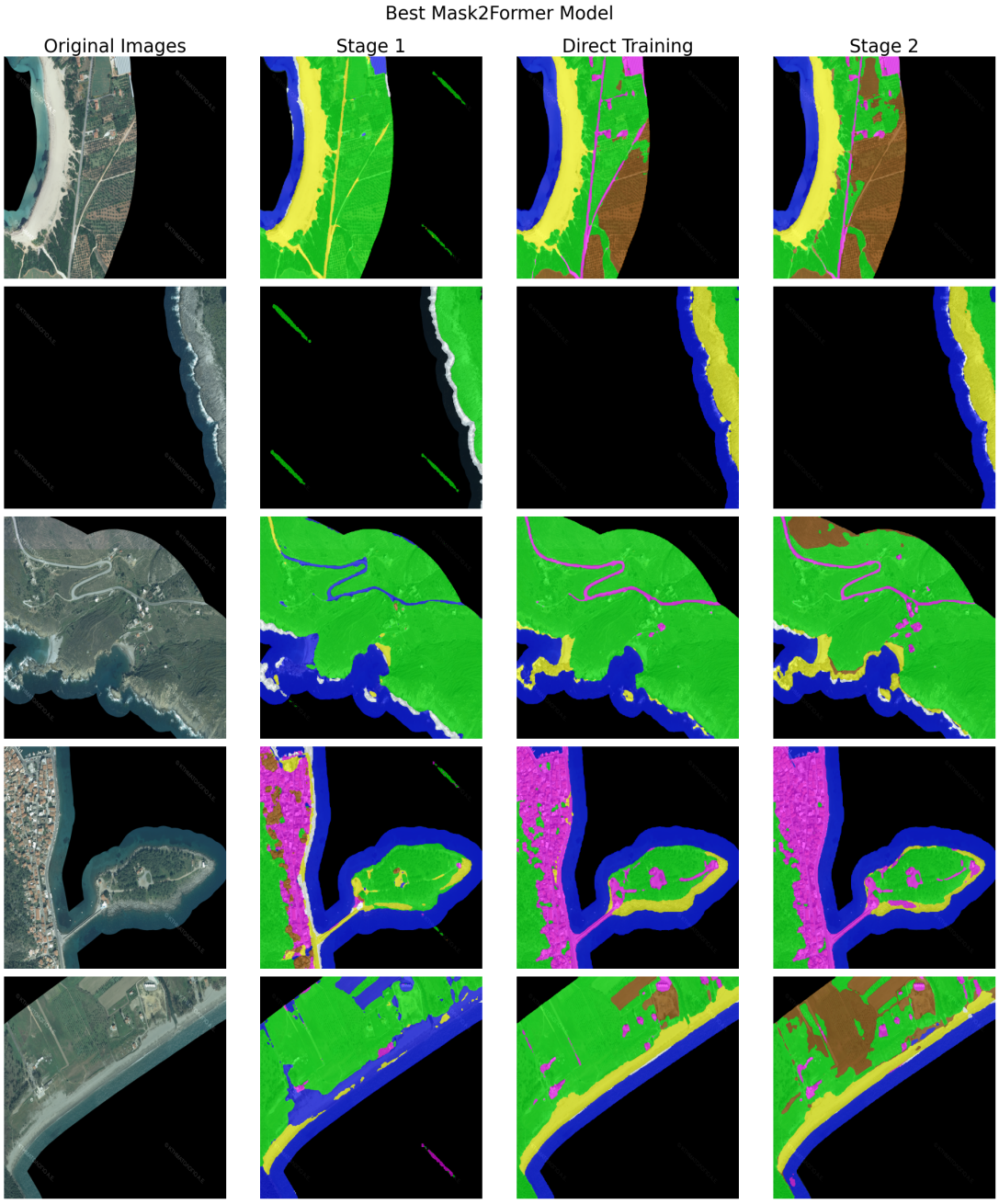


Figure 7.16: *Inference of best Mask2Former model on example images of the Greek Coastline dataset.*

7.4.4 Model Comparison (Stage 2)

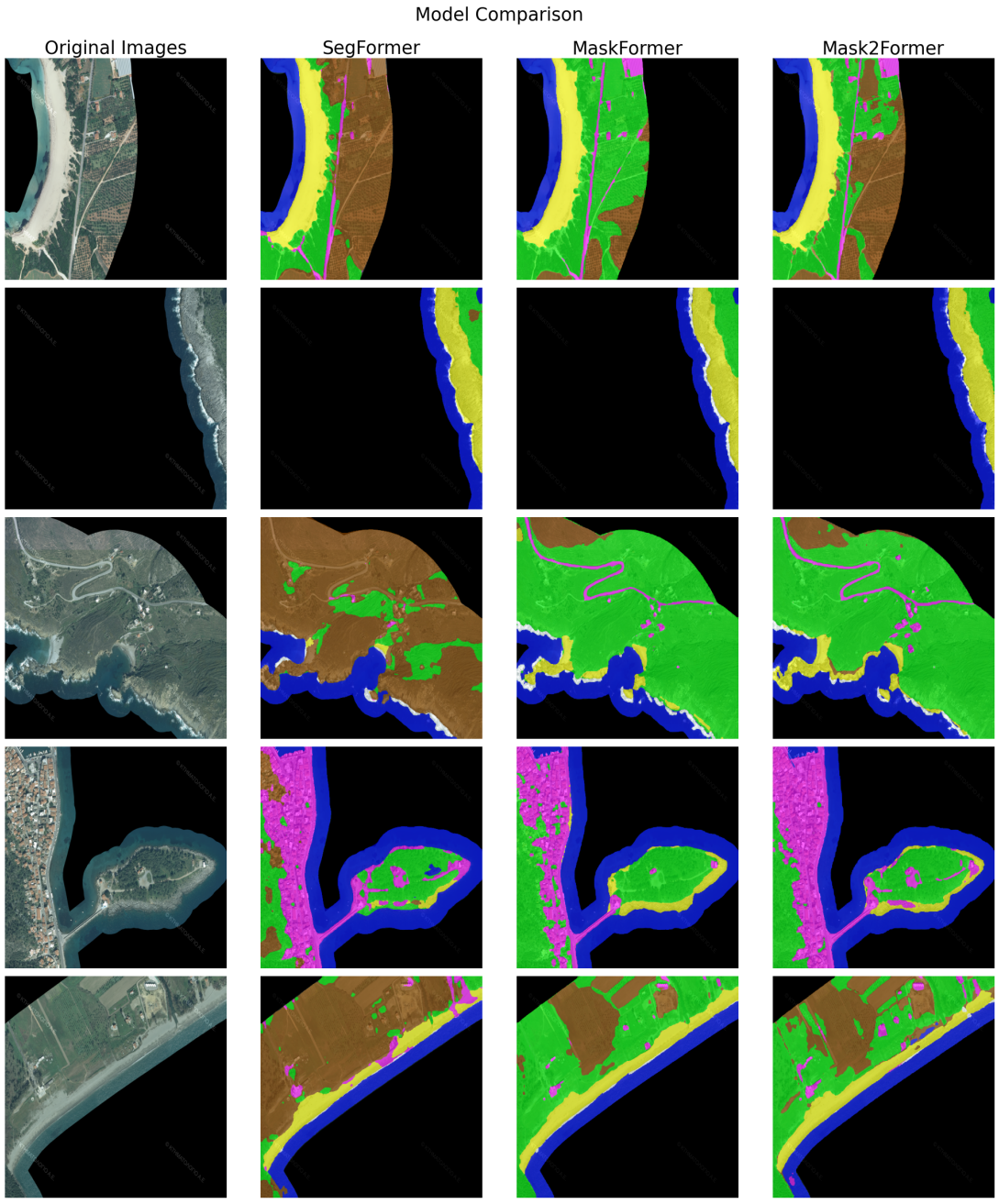


Figure 7.17: *Inference of the best stage 2 SegFormer, MaskFormer and Mask2Former models*

Chapter 8

Conclusion

8.1 Conclusion

In this diploma thesis we managed to perform semantic segmentation in remote sensing data of the Greek coastline using Transformer models. With this process we introduce useful information to the coastal images regarding their content, thus advancing the more general research in remote data analysis in coastal areas. Through the process of training the models as well as their visual evaluation, we concluded that Mask2Former had the best performance, achieving 85.43% mIoU on the grCoastline dataset. Additionally, we demonstrated the value of transfer learning as we chose to pre-train the models on a labeled dataset of images from the American coastline and then adapt them to our own dataset. Comparing the results of this process in relation to the direct training of the models on the Greek dataset, the superiority of the utilization of transfer learning is evident. Overall, we made an initial, yet vital, step towards the direction of leveraging computer vision techniques to further assist the ongoing in-situ research. In this thesis, we examined and preprocesed image datasets, we searched and studied SOTA models in the task of semantic segmentation and we succesfully trained them, gaining insightful results.

8.2 Future Work

As mentioned above, this work is the first and basic step in the context of a more general research concerning the geomorphological analysis of coastal environments through remote sensing data. Some of the future steps of this research are:

1. Introduce more classes for more detailed semantic segmentation of coastal images. For example, coasts could be separated into sandy or rocky, while development areas into houses, roads, harbors, etc. Most of these classes would provide useful and more universal information about the content of coastal areas.

2. Utilization of the third dimension that accompanies the images of the Greek Land Registry, that of the height of each pixel in relation to sea level. This height information could be particularly useful for slope analysis in certain areas.

3. The aerial images of the Land Registry have a resolution of 25cm, which is considered particularly high and can also be used for material level analysis. In particular, there is a need to extract information about the material of the beaches (type of sand, gravel, etc.), which until now has been carried out with in-situ measurements.

At the same time, the following can also be used to optimize our results:

1. Optimization of model training hyperparameters (learning rate, batch size, etc.).
2. Use more labeled data for training. Open Street Map could be used as a source.
3. Utilization of the 4th band of the photos (infrared) which can also provide useful information.
4. Training the OneFormer model [57] which is the new SOTA for image segmentation tasks.

Bibliography

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. *Attention is all you need*. *NeurIPS*, 2017.

[2] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. *Segformer: Simple and efficient design for semantic segmentation with transformers*. *NeurIPS*, 2021. DOI: https://doi.org/10.48550/arXiv.2105.15203 .

[3] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. *Per-pixel classification is not all you need for semantic segmentation*. *NeurIPS*, 2021. DOI: https://doi.org/10.48550/arXiv.2107.06278 .

[4] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, Rohit Girdhar. *Masked-attention Mask Transformer for Universal Image Segmentation*. *NeurIPS*, 2022.

[5] Vijay Badrinarayanan, Alex Kendall, Roberto Cipolla. *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. *IEEE*, 2016. DOI: https://doi.org/10.48550/arXiv.1511.00561 .

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. *Mask R-CNN*. 2018. DOI: https://doi.org/10.48550/arXiv.1703.06870 .

[7] Daan de Geus, Panagiotis Meletis, Gijs Dubbelman. *Fast Panoptic Segmentation Network*. 2019. DOI: https://doi.org/10.48550/arXiv.1910.03892 .

[8] *CS231n Convolutional Neural Networks for Visual Recognition*. https://cs231n.github.io/. (accessed: 07.02.2024).

[9] 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*. *International Conference on Learning Representations*, 2021.

[10] 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*. 2021. DOI: https://doi.org/10.48550/arXiv.2103.14030 .

[11] Salpea, N., Tzouveli, P., Kollias, D. *Medical Image Segmentation: A Review of Modern Architectures*. *Computer Vision - ECCV 2022 Workshops. ECCV 2022. Lecture Notes in Computer Science, vol 13807. Springer, Cham.*, 2023. DOI: https://doi.org/10.1007/978-3-031-25082-8_47 .

[12] G. Sapountzakis, P. -A. Theofilou and P. Tzouveli. *Covid-19 Detection From X-Rays Images Using Deep Learning Methods*. *IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW), Rhodes Island, Greece, 2023*, 2023. DOI: 10.1109/ICASSPW59220.2023.10193312 .

[13] Tzouveli, P., Simou, N., Stamou, G., Kollias, S. *Semantic classification of byzantine icons*. *IEEE Intelligent Systems 24, 35-43*, 2009.

[14] Lymperopoulos, E., Tzouveli P. Kollias,S. *Satellite image super-resolution for forest localization*. *International Conference on Machine Intelligence for GeoAnalytics and Remote Sensing (IEEE MIGARS), pp. 1-4*, 2023. DOI: 10.1109/MIGARS57353.2023.10064552 .

[15] Haralick, R. and Shapiro, L. *Computer and robot vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, United States, 1992.

[16] Weinland, D., Ronfard, R., and Boyer, E. *A survey of vision-based methods for action representation, segmentation and recognition*. *Computer Vision and Image Understanding*, 2011.

[17] Dalal, N. and Triggs, B. *Histograms of oriented gradients for human detection*. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[18] Lowe, D. G. *Distinctive Image Features from Scale-Invariant Keypoints*. *International Journal of Computer Vision*, 2004.

[19] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele. *The Cityscapes Dataset for Semantic Urban Scene Understanding*. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. DOI: https://doi.org/10.48550/arXiv.1604.01685 .

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, Li Fei-Fei. *ImageNet: A large-scale hierarchical image database*. *CVPR*, 2009. DOI: https://ieeexplore.ieee.org/document/5206848 .

[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, Piotr Dollár. *Microsoft COCO: Common Objects in Context*. 2014.

[22] E. H. Adelson. *On seeing stuff: the perception of materials by humans and machines*. IST/SPIE Electronic Imaging, 2001.

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso and Antonio Torralba. *Scene Parsing through ADE20K Dataset*. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[24] Martin Thoma. *A Survey of Semantic Segmentation*. 2016. DOI: https://doi.org/10.48550/arXiv.1602.06541 .

[25] Daniel Buscombe, Phillipe Wernette, Sharon Fitzpatrick, Jaycee Favela, Evan B. Goldstein, Nicholas M. Enwright. *A 1.2 Billion Pixel Human-Labeled Dataset for Data-Driven Classification of Coastal Environments*. *Scientific Data*, 2023. DOI: https://doi.org/10.1038/s41597-023-01929-2 .

[26] *Rasterio: access to geospatial raster data*. https://rasterio.readthedocs.io/en/stable/. (accessed: 20.01.2024).

[27] Drakopoulos P.G., E. Oikonomou, G. Skianis, S.E. Poulos, A.Vaiopoulos, K. Lazogiannis, G. Ghionis, A. Velegrakis. *Use of satellite imagery for automated monitoring of the shoreline retreat rate*. *Proceedings of the Adapt to Climate International Conference, Nicosia, Cyprus, pp.9*, 2014.

[28] Drakopoulos Panos, George Ghionis, Kostas. Lazogiannis, Serafim. Poulos. *Toward precise shoreline detection and extraction from remotely sensed images with the use of wet and dry sand spectral signatures*. *Fresenius environmental bulletin, 23, 2809-2813*, 2014.

[29] Rigos Anastasios, Aristidis Vaiopoulos, George Skianis, George Tsekouras, and Panos Drakopoulos. *A novel approach for automated shoreline extraction from remote sensing images using low level programming*. *Geophysical Research Abstracts, Vol. 17, EGU2015-7404, 2015 EGU General Assembly, Vienna Austria*, 2015.

[30] Kollias, Dimitrios and Arsenos, Anastasios and Kollias, Stefanos. *A deep neural architecture for harmonizing 3-D input data analysis and decision making in medical imaging*. *Neurocomputing*, 542:126244, 2023.

[31] Kollias, Dimitrios and Arsenos, Anastasios and Kollias, Stefanos. *Domain adaptation, explainability & fairness in ai for medical image analysis: Diagnosis of covid-19 based on 3-d chest ct-scans*. *arXiv preprint arXiv:2403.02192*, 2024.

[32] Hafeth, Deema Abdal and Kollias, Stefanos and Ghafoor, Mubeen. *Semantic Representations with Attention Networks for Boosting Image Captioning*. *IEEE Access*, 2023.

[33] Papaoikonomou, Antonios and Wingate, James and Verma, Vasudha and Durrant, Aiden and Ioannou, George and Papagiannis, Tasos and Yu, Miao and Alexandridis, Georgios and Dokhane, Abdelhamid and Leontidis, Georgios and others. *Deep learning techniques for in-core perturbation identification and localization of time-series nuclear plant measurements*. *Annals of Nuclear Energy*, 178:109373, 2022.

[34] Kollias, Stefanos and Yu, Miao and Wingate, James and Durrant, Aiden and Leontidis, Georgios and Alexandridis, Georgios and Stafylopatis, Andreas and Mylonakis, Antonios and Vinai, Paolo and Demaziere, Christophe. *Machine learning for analysis of real nuclear plant data in the frequency domain*. *Annals of Nuclear Energy*, 177:109293, 2022.

[35] Thota, Mamatha and Kollias, Stefanos and Swainson, Mark and Leontidis, Georgios. *Multi-source domain adaptation for quality control in retail food packaging*. *Computers in Industry*, 123:103293, 2020.

[36] Alhnaity, Bashar and Kollias, Stefanos and Leontidis, Georgios and Jiang, Shouyong and Schamp, Bert and Pearson, Simon. *An autoencoder wavelet based deep neural network with attention mechanism for multi-step prediction of plant growth*. *Information Sciences*, 560:35–50, 2021.

[37] Kollias, Dimitrios and Zafeiriou, Stefanos. *Exploiting multi-cnn features in cnn-rnn based dimensional emotion recognition on the omg in-the-wild dataset*. *IEEE Transactions on Affective Computing*, 12(3):595–606, 2020.

[38] Psaroudakis, Andreas and Kollias, Dimitrios. *MixAugment & Mixup: Augmentation Methods for Facial Expression Recognition*. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, σελίδες 2367–2375, 2022.

[39] De Sousa Ribeiro, Fabio and Calivá, Francesco and Swainson, Mark and Gudmundsson, Kjartan and Leontidis, Georgios and Kollias, Stefanos. *Deep bayesian self-training*. *Neural Computing and Applications*, 32(9):4275–4291, 2020.

[40] Ribeiro, Fabio De Sousa and Leontidis, Georgios and Kollias, Stefanos. *Capsule routing via variational bayes*. *Proceedings of the AAAI Conference on Artificial Intelligence*, τόμος 34, σελίδες 3749–3756, 2020.

[41] De Sousa Ribeiro, Fabio and Leontidis, Georgios and Kollias, Stefanos. *Introducing routing uncertainty in capsule networks*. *Advances in Neural Information Processing Systems*, 33:6490–6502, 2020.

[42] Kollias, Dimitrios and Yu, Miao and Tagaris, Athanasios and Leontidis, Georgios and Stafylopatis, Andreas and Kollias, Stefanos. *Adaptation and contextualization of deep neural network models*. *2017 IEEE symposium series on computational intelligence (SSCI)*, σελίδες 1–8. IEEE, 2017.

[43] Bouas, N and Vlaxos, Y and Brillakis, V and Seferis, M and Kollias, S. *Deep transparent prediction through latent representation analysis*. *arXiv preprint arXiv:2009.07044*, 2020.

[44] Robert A. Schowengerdt. *Remote sensing: models and methods for image processing*. ISBN 978-0-12-369407-2. Academic Press, 2007.

[45] Vikas Kookna. *Semantic vs. Instance vs. Panoptic Segmentation*. https://pyimagesearch.com/2022/06/29/semantic-vs-instance-vs-panoptic-segmentation/. (accessed: 20.01.2024).

[46] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, Piotr Dollár. *Panoptic Segmentation*. *CVPR*, 2019. DOI: https://doi.org/10.48550/arXiv.1801.00868 .

[47] Shruti Jadon. *A survey of loss functions for semantic segmentation*. *IEEE*, 2020. DOI: https://doi.org/10.1109/CIBCB48159.2020.9277638 .

[48] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[49] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. *Journal of Machine Learning Research*, 2014.

[50] Ian Goodfellow and Yoshua Bengio and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[51] Jonathan Long, Evan Shelhamer, Trevor Darrell. *Fully convolutional networks for semantic segmentation*. *IEEE*, 2015. DOI: https://ieeexplore.ieee.org/document/7298965 .

[52] Olaf Ronneberger, Philipp Fischer, Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. *MICCAI*, 2015. DOI: https://doi.org/10.48550/arXiv.1505.04597 .

[53] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, Alan L. Yuille. *DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs*. *TPAMI*, 2017. DOI: https://doi.org/10.48550/arXiv.1606.00915 .

[54] Philipp Krähenbühl, Vladlen Koltun. *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. *NIPS*, 2012. DOI: https://doi.org/10.48550/arXiv.1210.5644 .

[55] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, Hartwig Adam. *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. *ECCV*, 2018. DOI: https://doi.org/10.48550/arXiv.1802.02611 .

[56] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, Bin Xiao. *Deep High-Resolution Representation Learning for Visual Recognition*. *TPAMI*, 2020. DOI: https://doi.org/10.48550/arXiv.1908.07919 .

[57] Jitesh Jain, Jiachen Li, MangTik Chiu, Ali Hassani, Nikita Orlov, Humphrey Shi. *OneFormer: One Transformer to Rule Universal Image Segmentation*. *IEEE*, 2022. DOI: https://doi.org/10.48550/arXiv.2211.06220 .