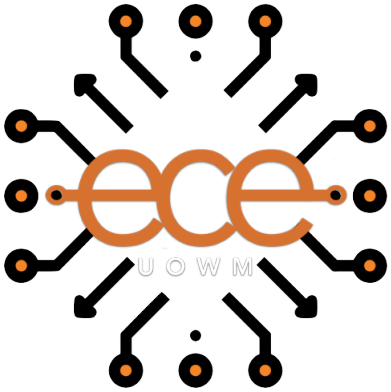


Πανεπιστήμιο Δυτικής Μακεδονίας

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

Αλγόριθμοι για την αναγνώριση αντικειμένων σε κρυπτογραφημένες εικόνες

Αλεξάνδρα Κυριζάκη (ΑΜ: 1494)

Επιβλέπων Καθηγητής: Νικόλαος Πλόσκας

11 Οκτωβρίου 2023

Περίληψη

Η μηχανική μάθηση και η υπολογιστική όραση αποτελούν τεχνολογίες αιχμής στην εποχή μας. Η χρήση αλγορίθμων εντοπισμού αντικειμένων σε στατικές εικόνες ή βίντεο στους τομείς αυτούς παρουσιάζει μια πλειάδα εφαρμογών στην ανθρώπινη καθημερινότητα και στην επιστημονική έρευνα, όπως η αναγνώριση ασθενειών από φωτογραφίες μαγνητικών και αξονικών τομογράφων, η χρήση αλγορίθμων μηχανικής μάθησης για την αναγνώριση αντικειμένων και ανθρώπων σε συστήματα παρακολούθησης, η διαλογή και συσκευασία φρούτων και λαχανικών που αναγνωρίστηκαν και ικανοποιούν προδιαγραφές εμφάνισης και ποιότητας και τα έξυπνα αυτοκίνητα που κινούνται ανεξάρτητα από την οδηγική ικανότητα του ανθρώπου την δεδομένη στιγμή. Αυτοί είναι κάποιοι από τους τομείς που μπορούν να αναφερθούν όπου εφαρμόζονται συστήματά μηχανικής μάθησης αναγνώρισης αντικείμενων με ιδιαίτερη επιτυχία. Ο ανθρώπινος παράγοντας βέβαια δεν πρέπει να υποτιμηθεί γιατί το ανθρώπινο μάτι έχει τη δυνατότητά να αναγνωρίζει αντικείμενα σε πραγματικό χρόνο. Όταν αυτά τα δεδομένα αφορούν στον προσωπικό χαρακτήρα του κάθε ανθρώπου, δεν θα πρέπει να διακινούνται ελεύθερα χωρίς τις απαραίτητες δικλείδες ασφαλείας. Αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης των εικόνων έρχονται να καλύψουν αποτελεσματικά αυτήν την απαίτηση. Σκοπός αυτής της εργασίας είναι η ανάλυση και η υπολογιστική μελέτη των αλγορίθμων μηχανικής μάθησης Faster R-CNN και YOLO για την ανίχνευση αντικειμένων σε εικόνες και των αλγορίθμων DES, AES και BlowFish που πραγματοποιούν την κρυπτογράφηση-αποκρυπτογράφηση στις εικόνες αυτές. Από την εκτέλεση των παραπάνω αλγορίθμων για την αναγνώριση αντικειμένων προέκυψε ότι ο αλγόριθμος YOLO παρουσιάζει καλύτερα αποτελέσματα σε σχέση με τον αλγόριθμο Faster R-CNN, τόσο στην ακρίβεια όσο και στον χρόνο εκτέλεσής του. Σχετικά με τη σύγκριση που έγινε ανάμεσα στους αλγορίθμους κρυπτογράφησης-αποκρυπτογράφησης εικόνων προέκυψε ότι ο αλγόριθμος AES είναι ο καταλληλότερος καθώς πραγματοποιεί την κρυπτο-

γράφηση σε λιγότερους γύρους και κατά συνέπεια αυτό τον κάνει τον πιο γρήγορο σε σχέση με τους αλγορίθμους DES και BlowFish.

Λέξεις κλειδιά: Ανίχνευση αντικειμένων, Νευρωνικά δίκτυα, Υπολογιστική όραση, Κρυπτογράφηση, Αποκρυπτογράφηση

Abstract

Machine learning and computer vision are cutting-edge technologies. The use of object detection algorithms in static images or videos in these fields has a multitude of applications in human daily life and scientific research, such as the identification of diseases from MRI and CT scan images, the use of machine learning algorithms to identify objects and people in surveillance systems, the sorting and packaging of fruit and vegetables that are identified and meet appearance and quality specifications, and smart cars that are driven independently of human driving ability at a given moment. These are just a few of the areas that can be mentioned where machine learning methods for object recognition have been applied with great success. The human factor should not be underestimated, of course, because the human eye has the ability to recognise objects in real time. When this data relates to the personal character of each individual, it should not be allowed to circulate freely without the necessary security. Algorithms for encrypting and decrypting images come to effectively meet this requirement. The purpose of this paper is to analyze and computationally study the Faster R-CNN and YOLO machine learning algorithms for object detection in images and the DES, AES, and BlowFish algorithms that perform encryption-decryption on these images. From the execution of the above algorithms for object recognition it was found that the YOLO algorithm shows better results than the Faster R-CNN algorithm both in accuracy and execution time. Regarding the comparison made between the image encryption-decryption algorithms, it was found that the AES algorithm is the most suitable one as it performs the encryption in fewer rounds and this makes it the fastest compared to the DES and BlowFish algorithms.

Keywords: Object detection, Neural networks, Computer vision, Encryption, Decryption

Δήλωση Πνευματικών Δικαιωμάτων

Δήλωση Πνευματικών Δικαιωμάτων Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν. 1599/1986 και τα άρθρα 2,4,6 παρ. 3 του Ν. 1256/1982, η παρούσα Διπλωματική Εργασία με τίτλο ”Αλγόριθμοι για την αναγνώριση αντικειμένων σε κρυπτογραφημένες εικόνες” καθώς και τα ηλεκτρονικά αρχεία και πηγαίοι κώδικες που αναπτύχθηκαν ή τροποποιήθηκαν στα πλαίσια αυτής της εργασίας και αναφέρονται ρητώς μέσα στο κείμενο που συνοδεύουν, και η οποία έχει εκπονηθεί στο Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Πανεπιστημίου Δυτικής Μακεδονίας, υπό την επίβλεψη του μέλους του Τμήματος κ. Νικολάου Πλόσκα αποτελεί αποκλειστικά προϊόν προσωπικής εργασίας και δεν προσβάλλει κάθε μορφής πνευματικά δικαιώματα τρίτων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον. Τα σημεία όπου έχω χρησιμοποιήσει ιδέες, κείμενο, αρχεία ή / και πηγές άλλων συγγραφέων, αναφέρονται ευδιάκριτα στο κείμενο με την κατάλληλη παραπομπή και η σχετική αναφορά περιλαμβάνεται στο τμήμα των βιβλιογραφικών αναφορών με πλήρη περιγραφή.

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

Υπογραφή Φοιτητή

Περιεχόμενα

Κατάλογος σχημάτων

Κατάλογος πινάκων

Κατάλογος απεικονίσεων

Κεφάλαιο 1

Εισαγωγή

1.1 Ορισμός του προβλήματος

Ο εντοπισμός και η αναγνώριση αντικειμένων σε εικόνες και βίντεο επιτυγχάνεται χρησιμοποιώντας αλγόριθμους Νευρωνικών Δικτύων. Η ταχύτητα επεξεργασίας και εντοπισμού των αντικειμένων με τη χρήση ισχυρών υπολογιστικών μηχανών και με τη συμβολή από εξίσου ισχυρές κάρτες γραφικών μας δίνει τη δυνατότητα αναγνώρισης αντικειμένων ακόμη και σε πραγματικό χρόνο. Ένα αυτοκινούμενο όχημα στο οποίο μια τέτοια τεχνολογία θα συμβάλει στην αποφυγή ατυχημάτων, μειώνοντας τον επιρρεπή πολλές φορές ανθρώπινο παράγοντα, τη στιγμή της κρίσιμης απόφασης κατά την οδήγηση. Αλγόριθμοι αναγνώρισης αντικειμένων εγκατεστημένοι σε γεωργικά μηχανήματα φυτοπροστασίας, μας επιτρέπουν την ανίχνευση της παθογένειας του φυτού και τον ψεκασμό του με το κατάλληλο φυτοφάρμακο στην απαραίτητη ποσότητα. Όταν αυτά τα δεδομένα είναι προσωπικού χαρακτήρα η χρήση και ο χειρισμός τους θα πρέπει να είναι σύμφωνος με τον Γενικό Κανονισμό Προστασίας Προσωπικών Δεδομένων (GDPR). Η χρήση αλγορίθμων που τα κρυπτογραφούν-αποκρυπτογραφούν συμβάλει στην διαφύλαξή τους.

1.2 Κίνητρα και Στόχοι Υλοποίησης

Ο στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η μελέτη των αλγορίθμων που χρησιμοποιούνται για την αναγνώριση αντικειμένων, καθώς επίσης και των αλγορίθμων κρυπτογράφησης που μπορούν να εφαρμοστούν στα δεδομένα που περιέχουν αυτά τα αντικείμενα. Από τους αλγορίθμους που μελετήθηκαν, επιλέχθηκαν αυτοί που έχουν την μέγιστη δυνατή απόδοση εξαγωγής των αποτελεσμάτων

που επιθυμούμε στον καλύτερο δυνατό χρόνο. Πραγματοποιήθηκαν δύο ξεχωριστές υπολογιστικές μελέτες που αφορούσαν στο κάθε είδος αλγορίθμου.Στην υπολογιστική μελέτη των αλγορίθμων αναγνώρισης εικόνων εξετάστηκαν τα χαρακτηριστικά δύο συγκεκριμένων αλγορίθμων που εξυπηρετούσαν αυτόν τον σκοπό, για τους οποίους επιλέχθηκαν τα κατάλληλα μοντέλα εκμάθησης. Στην υπολογιστική μελέτη των αλγορίθμων κρυπτογράφησης έγιναν μετρήσεις σε τρεις συμμετρικούς αλγόριθμους μεταβάλλοντας το μήκος κλειδιού. Τα δεδομένα των μετρήσεων για όλους τους αλγορίθμους και των δύο μελετών συγκρίθηκαν μεταξύ τους ώστε να εξαχθούν ουσιαστικά συμπεράσματα για τη χρήση τους.

1.3 Διάρθρωση κειμένου

Η διάρθρωση της συγκεκριμένης διπλωματικής εργασίας είναι η εξής: Στο Κεφάλαιο 2 αρχικά έγινε μια σύντομη περιγραφή των εννοιών της αναγνώρισης αντικειμένων και των νευρωνικών δικτύων και στη συνέχεια έγινε παρουσίαση και ανάλυση των αλγορίθμων αναγνώρισης αντικειμένων R-CNN, Fast R-CNN, Faster R-CNN, R-FCN, HOG, SSD, SPP-net και YOLO και πραγματοποιήθηκε μια θεωρητική σύγκριση για αυτούς. Τέλος, δόθηκαν ορισμοί για τις έννοιες συμμετρική και ασύμμετρη κρυπτογράφηση καθώς επίσης και ανάλυση των αλγορίθμων που ανήκουν σε αυτές τις δυο κατηγορίες DES, AES, BlowFish, RSA, DDS, ElGamal και μια σύγκριση για αυτούς. Στο Κεφάλαιο 3 έγινε μια έρευνα σε διάφορες μελέτες επιστημών σχετικά με το ποιοι αλγόριθμοι αναγνώρισης αντικειμένων και κρυπτογράφησης-αποκρυπτογράφησης έχουν χρησιμοποιηθεί και τι αντικείμενα έθεταν προς εξέταση οι εκάστοτε μελετητές σε καθεμία από αυτές. Το Κεφάλαιο 4 πραγματεύεται τον τρόπο υλοποίησης σε θεωρητικό επίπεδο των αλγορίθμων Faster R-CNN, YOLO, DES, AES και Blowfish και περιέχει και τους αντίστοιχους ψευδοκώδικες για καθένα από αυτούς. Έπειτα το Κεφάλαιο 5 περιέχει την υπολογιστική μελέτη των αλγορίθμων που αναλύθηκαν στο προηγούμενο κεφάλαιο και παρουσιάζονται τα αποτελέσματα των μετρήσεων που πραγματοποιήθηκαν για αυτούς σε διάφορα σύνολα δεδομένων φωτογραφιών ίδιων και διαφορετικών αντικειμένων. Τέλος, στο Κεφάλαιο 6 παραθέτονται τα συμπεράσματα της συγκεκριμένης διπλωματικής εργασίας και επιπλέον αναφέρονται οι τομείς και οι εφαρμογές στους οποίους βρίσκει χρηστικότητα.

Κεφάλαιο 2

Ορισμοί και Έννοιες

2.1 Αλγόριθμοι Αναγνώρισης Αντικειμένων

Η αναγνώριση αντικειμένων είναι η ανίχνευση αντικειμένων από μία ή περισσότερες κλάσεις σε μια εικόνα, δηλαδή με βάση χαρακτηριστικά που εμφανίζονται στην εικόνα οι αλγόριθμοι έχουν την ικανότητα να κάνουνε συγκρίσεις με άλλα αντικείμενα που βρίσκονται σε διάφορες κατηγορίες και να αντιλαμβάνονται σε ποια κατηγορία (κλάση) βρίσκεται το αντικείμενο που απεικονίζεται τη συγκεκριμένη χρονική στιγμή μπροστά μας. Κάθε ανίχνευση αντικειμένου εξαρτάται από τη μορφή στην οποία βρίσκεται η πληροφορία, δηλαδή από χαρακτηριστικά όπως η θέση, η κλίμακα και το πλαίσιο οριοθέτησης του αντικειμένου καθώς επίσης και από την ύπαρξη ή μη μιας μάσκας τμηματοποίησης του αντικειμένου. Αναλόγως με το αντικείμενο το οποίο επιθυμούμε να αναγνωριστεί υπάρχουν και άλλοι πιο λεπτομερείς παράγοντες που επηρεάζουν την ταυτοποίηση του όπως είναι οι γραμμικοί ή οι μη γραμμικοί μετασχηματισμοί. Για παράδειγμα στην ανίχνευση προσώπων οι αλγόριθμοι υπολογίζουν τη θέση των βασικών χαρακτηριστικών που είναι τα μάτια, η μύτη και το στόμα.

Η αναγνώριση αντικειμένων βρίσκει εφαρμογή σε πολλούς τομείς, όπως στα συστήματα ανίχνευσης ασφαλείας, στην αυτόνομη οδήγηση, στην υγειονομική περίθαλψη, στο λιανεμπόριο, στη μεταποίηση προϊόντων και στη γεωργία. Πιο συγκεκριμένα χρησιμοποιείται για την ανίχνευση ατόμων, οχημάτων και ζώων σε ένα βίντεο ή μια εικόνα, στην ανίχνευση προσώπων, την καταμέτρηση ατόμων σε έναν χώρο, την ανίχνευση κειμένου είτε αυτό βρίσκεται μέσα σε μια εικόνα είτε αποτελεί χειρόγραφο κείμενο κάποιου που έχει ψηφιοποιηθεί, στην αυτοκινητοβιομηχανία έτσι

ώστε να οχήματα να μπορούν να αναγνωρίζουν ορθά τα αντικείμενα που βρίσκονται γύρω τους στον χώρο και να παρέχουν στον οδηγό-χρήστη τις απαραίτητες συμβουλές για πιο ασφαλή και ξεκούραστη οδήγηση και τέλος στον τομέα του συστήματος υγείας χρησιμοποιείται για την αναγνώριση ασθενειών και στην παροχή πιο εξειδικευμένων θεραπειών και λύσεων στον ασθενή από τον γιατρό.

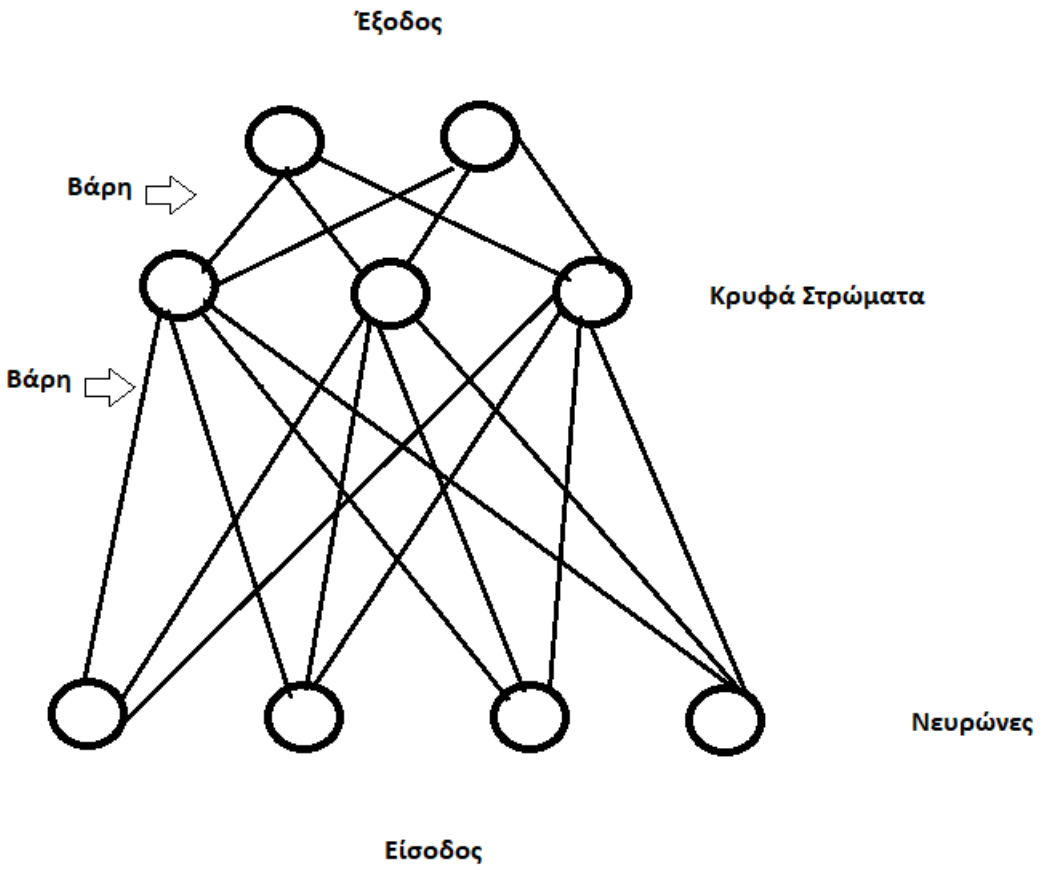
Από όλα αυτά μπορούμε να συμπεράνουμε ότι η αναγνώριση αντικειμένων έχει ένα ευρύ φάσμα εφαρμογών που καλύπτει αντικείμενα όπως την διευκόλυνση και βελτιστοποίηση καθημερινών διαδικασιών και χρήση μηχανών με περισσότερη ασφάλεια. Ανάλογα με τη φύση των δεδομένων μπορούν να χρησιμοποιηθούν διαφορετικές τεχνικές για την αναγνώριση αντικειμένων. Μια από αυτές τις τεχνικές είναι τα νευρωνικά δίκτυα.

2.1.1 Νευρωνικά Δίκτυα

Ένα τεχνητό νευρωνικό δίκτυο, που πιο συχνά συναντάται με την ονομασία νευρωνικό δίκτυο, αποτελείται από τρία στρώματα νευρώνων τα οποία διαχωρίζονται στης εξής κατηγορίες: ένα στρώμα νευρώνων εισόδου (ή κόμβων, μονάδων), ένα ή περισσότερα κρυφά στρώματα νευρώνων και ένα στρώμα νευρώνων εξόδου. Στο Σχήμα 2.1 απεικονίζεται η αρχιτεκτονική του νευρωνικού δικτύου και οι συνδέσεις μεταξύ των νευρώνων. Κάθε σύνδεση έχει ένα βάρος, το οποίο είναι μια αριθμητική τιμή. Η έξοδος του νευρώνα i του κρυμμένου στρώματος h_i, υπολογίζεται από την εξής σχέση:

$$h_i = s\left(\sum_{j=1}^{N} \nu_{ij}\chi_j + \top_i^{h_i d}\right) \tag{2.1}$$

όπου $s()$ είναι η συνάρτηση ενεργοποίησης (ή μεταφοράς), ο αριθμός των εισόδων νευρώνων, Vij τα βάρη, xj οι είσοδοι στους νευρώνες εισόδου, και $Tihid$ το κατώφλι των κρυφών νευρώνων. Εκτός από την προσθήκη μη γραμμικότητας στο νευρωνικό δίκτυο, η παραπάνω συνάρτηση αποσκοπεί στον περιορισμό της τιμής του νευρώνα ώστε να αποτρέψει την παράλυση του νευρωνικού δικτύου από αποκλίνοντες νευρώνες [36]. Τα νευρωνικά δίκτυα έχουν πολλές χρήσεις και εφαρμογές σε πολλούς τομείς, ένας από αυτούς είναι και η αναγνώριση εικόνων, όπως η ταξινόμηση ιατρικών εικόνων, η όραση υπολογιστών και η αναγνώριση αντικειμένων [12].



Σχήμα 2.1: Αρχιτεκτονική νευρωνικού δικτύου

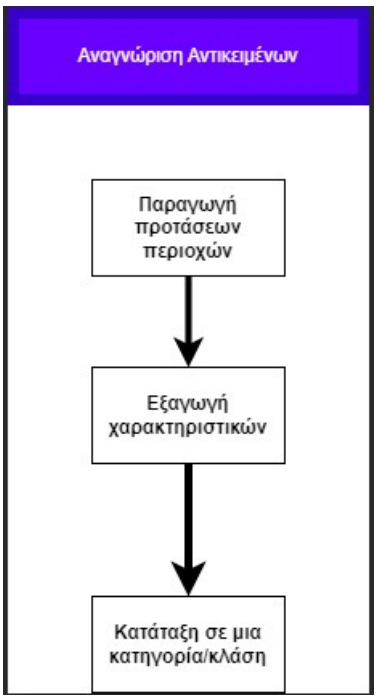
2.1.2 Region-based Convolutional Neural Network (R-CNN)

Ο R-CNN είναι ένα συνελικτικό δίκτυο που αναπτύχθηκε από μια ομάδα ερευνητών του UC Berkely το 2014 [11]. Έχει την ικανότητα να ανιχνεύει διαφορετικούς τύπους αντικειμένων σε μια εικόνα. Η μόνη διαφορά που παρατηρείται στον αλγόριθμο αυτό σε σχέση με την προαναφερθείσα μέθοδο αναγνώρισης αντικειμένων (Σχήμα 2.2) είναι ότι εξάγει τα χαρακτηριστικά με βάση ένα συνελικτικό δίκτυο (CNN). Η λειτουργία του R-CNN μπορεί να χωριστεί σε τρία βήματα τα οποία είναι τα ακόλουθα και φαίνονται και στο Σχήμα 2.3.

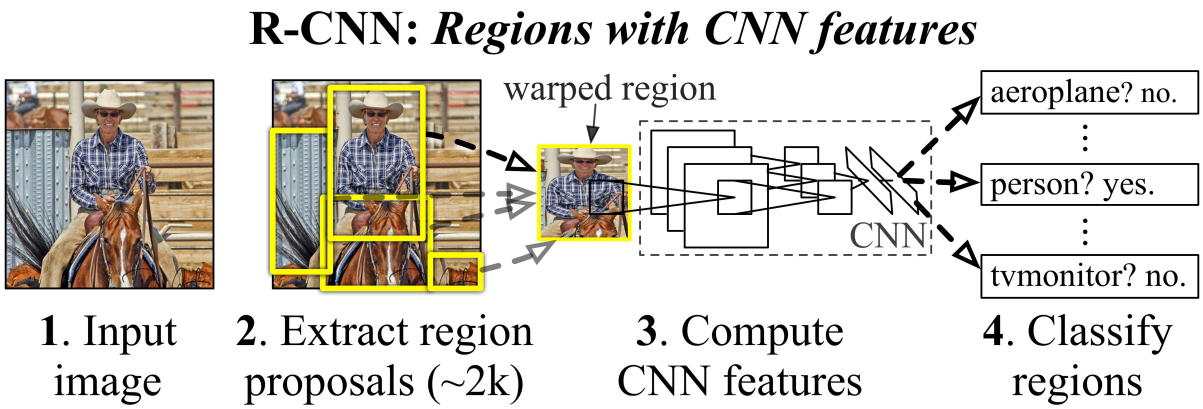
- Χρησιμοποιώντας τον αλγόριθμο επιλεκτικής αναζήτησης παράγει 2,000 προτάσεις περιοχών.
- Αλλάζει και επαναπροσδιορίζει το μέγεθος του αντικειμένου ως ένα σταθερό προκαθορισμένο μέγεθος, και στη συνέχεια πραγματοποιείται το δεύτερο βήμα του αλγορίθμου που είναι η εξαγωγή διανύσματος χαρακτηριστικών από κάθε πρόταση περιοχής με μήκος 4,096.
- Τέλος, χρησιμοποιεί τον προ-εκπαιδευμένο αλγόριθμο SVM για να κάνει ταξινόμησή τις προτάσεις περιοχής είτε με βάσει το φόντο τους είτε κάνοντας χρήση κάποια από τις ήδη υπάρχουσες κλάσεις αντικειμένων.

Η λειτουργία του R-CNN έχει και κάποια μειονεκτήματα τα οποία είναι τα ακόλουθα:

- Δεν μπορεί να γίνει εκπαίδευση από άκρη σε άκρη διότι ο αλγόριθμος R-CNN είναι ένα μοντέλο πολλαπλών σταδίων που το καθένα έχει ανεξάρτητη συνιστώσα.
- Ο R-CNN δεν μπορεί να εκτελεστεί σε πραγματικό χρόνο καθώς κάθε πρόταση περιοχής τροφοδοτείται με τρόπο ανεξάρτητο για την εξαγωγή των χαρακτηριστικών.Επιπλέον επειδή εξαρτάται για τη δημιουργία των προτάσεων περιοχών από τον αλγόριθμο επιλεκτικής αναζήτησης, είναι χρονοβόρος.
- Απαιτεί εκατοντάδες gigabytes αποθηκευτικού χώρου διότι ο αλγόριθμος R-CNN αποθηκεύει στον δίσκο τα εξαγόμενα χαρακτηριστικά από τον εκπαιδευμένο CNN έτσι ώστε στη συνέχεια να εκπαιδεύσει το SVM [9].



Σχήμα 2.2: Ροή για τον τρόπο αναγνώρισης αντικείμενων



Σχήμα 2.3: Παράδειγμα R-CNN [9]

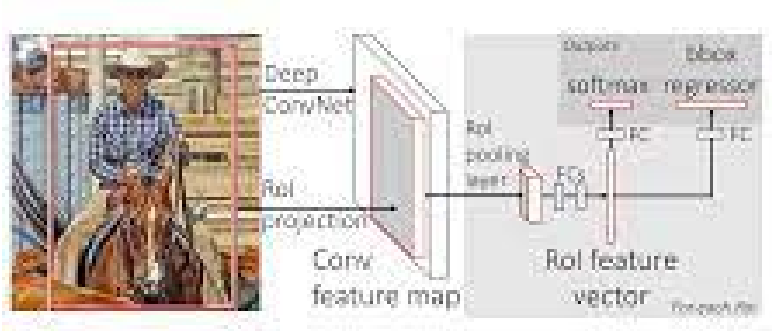
2.1.3 Fast R-CNN

Ο Fast R-CNN είναι ένας αλγόριθμος αναγνώρισης αντικειμένων που αναπτύχθηκε από τον Ross Girshick[10]. Παρουσιάζει βελτιώσεις σε σχέση με τον R-CNN στην ταχύτητα του. Ο αλγόριθμος αυτός έχει τα εξής χαρακτηριστικά:

- Εξάγει από όλες τις προτάσεις στην ίδια εικόνα διανύσματα χαρακτηριστικών ίσου μήκους χρησιμοποιώντας ένα νέο στρώμα που ονομάζεται ROI Pooling.
- Δημιουργεί ένα δίκτυο που έχει μόνο ένα στάδιο σε σύγκριση με το τον R-CNN που έχει πολλαπλά.
- Είναι πιο ακριβής συγκριτικά με τον απλό R-CNN.
- Είναι ταχύτερος σε σχέση με τον απλό R-CNN.
- Χρειάζεται λιγότερο χώρο στον δίσκο σε σχέση με τον R-CNN διότι δεν αποθηκεύει στην προσωρινή μνήμη τα εξαγόμενα χαρακτηριστικά.

Η αρχιτεκτονική του μοντέλου αυτού του αλγορίθμου, όπως φαίνεται και στο Σχήμα 2.4, αποτελείται μόνο από ένα στάδιο που λειτουργεί ως εξής, δέχεται μια εικόνα ως είσοδο και επιστρέφει τις πιθανότητες κάθε κλάσης και τα πλαίσια οριοθέτησης των αντικειμένων που εντοπίστηκαν. Πιο αναλυτικά, ο χάρτης των χαρακτηριστικών από το τελευταίο συνελικτικό στρώμα τροφοδοτεί ένα ROI Pooling layer το οποίο λειτουργεί κάνοντας διαχωρισμό κάθε πρότασης περιοχής σε ένα πλέγμα κελιών, και σε κάθε κελί εφαρμόζεται για να επιστρέψει μια ενιαία τιμή η λειτουργία max pooling. Αυτό πραγματοποιείται για να εξαχθεί από κάθε πρόταση περιοχής ένα διάνυσμα χαρακτηριστικών, που είναι το σύνολο των τιμών από όλα τα κελιά και το μήκος του εξαρτάται από το μέγεθος του πλέγματος, με σταθερό μήκος. Στη συνέχεια το διάνυσμα που εξάγεται κατά την χρήση του ROI Pooling μεταφέρεται σε ορισμένα στρώματα FC και η έξοδος του τελευταίου στρώματος χωρίζεται σε δύο κλάδους που είναι οι ακόλουθοι:

- Softmax layer που προβλέπει τη βαθμολογία της κλάσης
- FC layer που προβλέπει τα πλαίσια οριοθέτησης των ανιχνευμένων αντικειμένων



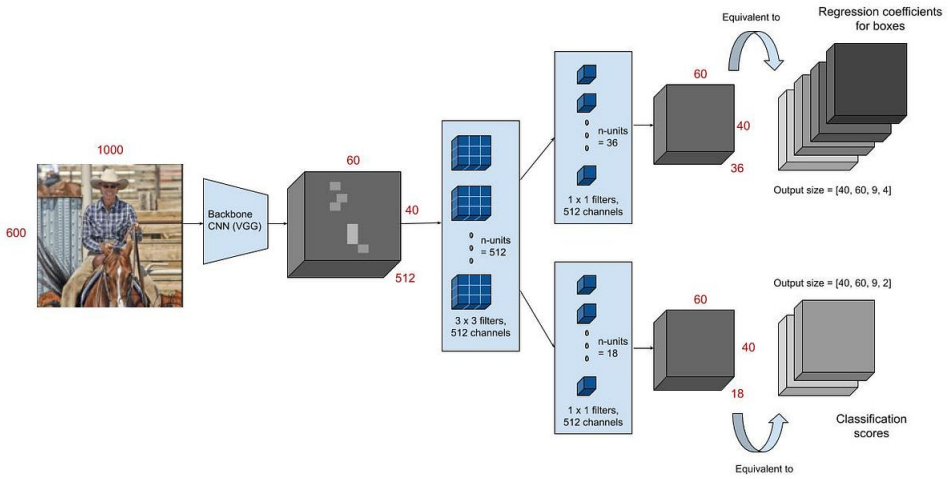
Σχήμα 2.4: Παράδειγμα Fast R-CNN [9]

Ο Fast R-CNN μοιράζει τους υπολογισμούς για την εύρεση των χαρακτηριστικών σε πολλαπλές προτάσεις και έτσι επιτυγχάνεται και η ταχύτητα του σε σχέση με τον R-CNN που κάθε πρόταση περιοχής τροφοδοτείται στο μοντέλο ανεξάρτητα από τις άλλες προτάσεις περιοχής. Ο αλγόριθμος Fast R-CNN πέρα από τα χαρακτηριστικά που τον καθιστούν κατάλληλο για αναγνώρισή εικόνων έχει ένα κρίσιμο μειονέκτημα το οποίο προκύπτει λόγω της εξάρτησής του από τον χρονοβόρο αλγόριθμο Selective Search που χρησιμοποιείται για τη δημιουργία προτάσεων περιοχών. Αυτή η επιλεκτική μέθοδος αναζήτησής ενδέχεται να μην είναι αρκετά ακριβής για την ανίχνευση όλων των αντικειμένων-στόχων στο σύνολο δεδομένων, καθώς δεν μπορεί να προσαρμοστεί σε μια συγκεκριμένη εργασία ανίχνευσης αντικειμένων [9].

2.1.4 Faster R-CNN

Ο αλγόριθμος Faster R-CNN για την παραγωγή προτάσεων περιοχών δημιουργεί ένα δίκτυο. Ο αλγόριθμος αυτός αποτελεί την επέκταση του αλγορίθμου Fast R-CNN και είναι ταχύτερος εξαιτίας του δικτύου προτάσεων περιοχής (RPN), και λειτουργεί με τον παρακάτω τρόπο:

- Το δίκτυο RPN δημιουργεί προτάσεις περιοχών.
- Χρησιμοποιεί ένα ROI Pooling Layer για να εξάγει ένα διάνυσμα χαρακτηριστικών σταθερού μήκους από κάθε πρόταση περιοχής μιας εικόνας.



Σχήμα 2.5: Παράδειγμα Faster R-CNN [9]

- Τα εξαγόμενα διανύσματα χαρακτηριστικών στη συνέχεια ταξινομούνται χρησιμοποιώντας τον αλγόριθμο Fast R-CNN.
- Τέλος, επιστρέφονται τα σκορ των κλάσεων από τα αντικείμενα που ανιχνεύτηκαν μαζί με τα πλαίσια οριοθέτησης.

Η σχηματική απεικόνιση του αλγορίθμου παρουσιάζεται παρακάτω στο Σχήμα 2.5.

Η αρχιτεκτονική του Faster R-CNN αποτελείται από δύο ενότητες:

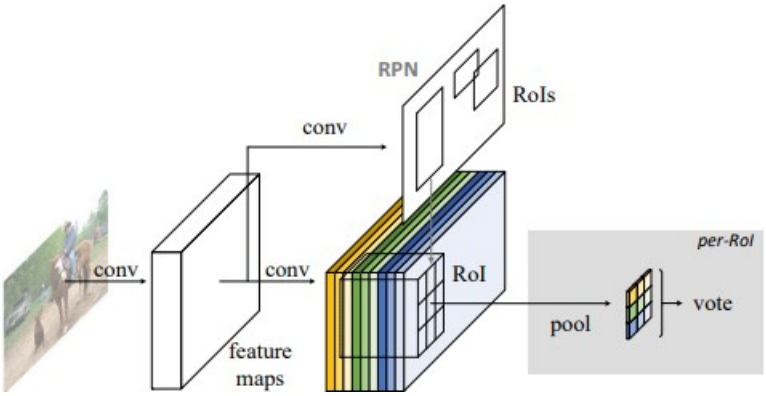
- RPN: Που παράγει τις προτάσεις περιοχών.
- Fast R-CNN: Που χρησιμοποιείται για την ανίχνευση αντικειμένων σε κάθε πρόταση περιοχής.

Η ενότητα του RPN είναι υπεύθυνη για την παραγωγή των προτάσεων περιοχών και καθοδηγεί την ενότητα ανίχνευσης του Fast R-CNN , για το που να αναζητήσει για αντικείμενα σε μια εικόνα, με εφαρμογή της έννοιά της προσοχής στα νευρωνικά δίκτυα. Η ενότητα του Fast R-CNN λειτουργεί όπως περιγράφηκε στην αντίστοιχη Ενότητα Fast R-CNN. Ο Faster R-CNN έχει τόσο θετικά όσο και αρνητικά χαρακτηριστικά. Ένα θετικό του αλγορίθμου αυτού είναι ότι επιτυγχάνει

καλό αποτέλεσμα στο σύνολο δεδομένων κοινής ανίχνευσης. Επιτυγχάνει την κορυφαία ακρίβεια ανίχνευσης αντικειμένων για το μοντέλο VGG-16 στο Pascal VOC 2007 και PASCAL VOC 2012 με ποσοστά 73.2% και 70.4%, αντίστοιχα. Παρόλο τα θετικά που έχει στο τομέα της ακρίβειας ο αλγόριθμος Faster R-CNN παρουσιάζει μη ιδανικά αποτελέσματα στην ανίχνευση μικρών αντικειμένων. Αυτό οφείλεται στο γεγονός ότι παρουσιάζει κορυφαίες επιδόσεις στα σύνολα δεδομένων PASCAL VOC και ανιχνεύει αντικείμενα που καταλαμβάνουν συνήθως την πλειονότητα μιας εικόνας. Αυτό έχει ως αποτέλεσμα, μικρά αντικείμενα χαμηλής ανάλυσης να είναι δύσκολα να εντοπιστούν από τον αλγόριθμο στην εικόνα. Επιπλέον, αυτή η αστοχία οφείλεται και στο γεγονός ότι το ROI-pooling layer δημιουργεί χαρακτηριστικά για την ανίχνευση μόνο από τον τελευταίο χάρτη χαρακτηριστικών, με αποτέλεσμα ο ανιχνευτής να παρουσιάζει δυσκολία στην πρόβλεψη της κλάσης του αντικειμένου και του πλαισίου οριοθέτησης. Ένα δεύτερο μειονέκτημα είναι ότι οι εικόνες SAR είναι αρκετά διαφορετικές από τις κοινές οπτικές εικόνες. Οπότε η χρησιμοποίηση των προ-εκπαιδευμένων επιπέδων στην ανίχνευση SAR εικόνων δεν θα έχει καλή απόδοση [9, 21].

2.1.5 Region-based Fully Convolutional Network (R-FCN)

Ο αλγόριθμος R-FCN είναι ένας πλήρως συνελικτικός αλγόριθμος και σχεδόν όλοι οι υπολογισμοί για την ανίχνευση αντικειμένων μοιράζονται σε ολόκληρη την εικόνα. Για να ενσωματώσουμε τη μεταφραστική διακύμανση στο FCN δημιουργούνται χάρτες βαθμολογίας που είναι ευαίσθητοι στη θέση και χρησιμοποιούνται για την επίλυση του διλήμματος μεταξύ της μεταφραστικής αναλλοίωτης ταξινόμησης εικόνας και μεταφραστικής διακύμανσης στην ανίχνευση αντικειμένων. Οι χάρτες αυτοί κατασκευάζονται χρησιμοποιώντας εξειδικευμένα επίπεδα συνενελίξεων ως έξοδο του FCN. Το σύνολο αυτών των συνελίξεων συγκεντρώνονται σε μια τράπεζα. Η λειτουργία αυτών των χαρτών είναι να κωδικοποιούν την πληροφορία θέσης σε σχέση με μια άλλη σχετική χωρική θέση στην εικόνα. Στη συνέχεια πάνω σε αυτή την FCN έξοδο εφαρμόζουμε ένα ευαίσθητο στη θέση ROI pooling layer που εξάγει πληροφορίες από τους χάρτες αυτούς χωρίς να προσθέτει βάρη. Ολόκληρη η αρχιτεκτονική του R-FCN μαθαίνεται από άκρη σε άκρη δηλαδή όλα τα συνελικτικά στρώματα μοιράζονται σε ολόκληρη την εικόνα αλλά κωδικοποιούν χωρικές πλη-



Σχήμα 2.6: Παράδειγμα R-FCN [5]

ροφορίες που είναι απαραίτητες για την ανίχνευση αντικειμένων. Στο Σχήμα 2.6 απεικονίζεται ένας $kxk = 3x3$ ευαίσθητος στη θέση χάρτης πεδίου εφαρμογής που παράγεται από ένα πλήρως συνελικτικό δίκτυο. Ο αλγόριθμος R-FCN είναι 2.5-20 φορές ποιο γρήγορος σε σχέση με τον Faster R-CCN[5].

Ο αλγόριθμος R-FCN έχει τα εξής πλεονεκτήματα:

- Με τον υπολογισμό των χαρτών χαρακτηριστικών με βάση την περιοχή, οι οποίοι είναι ανεξάρτητοι από το ROI pooling layer, μειώνεται ο όγκος εργασίας που απαιτείται για κάθε περιοχή ενδιαφέροντος.
- Επιτυγχάνει ισορροπία μεταξύ δυο σημαντικών χαρακτηριστικών που είναι η ακρίβεια και η ταχύτητα χρησιμοποιώντας τους χάρτες πεδίου εφαρμογής.

Και βέβαια έχει και κάποια μειονεκτήματά τα οποία είναι τα εξής:

- Ο αλγόριθμος R-FCN είναι λιγότερο ακριβής στα αποτελέσματα που εξάγει σε σχέση με τον αλγόριθμο Faster R-CNN.
- Απαιτεί μεταγενέστερη επεξεργασία όπως είναι η μη μέγιστη καταστολή που είναι η πιο χρονοβόρα διαδικασία για τα ταχύτερα μοντέλα.

2.1.6 Histogram of Oriented Gradients (HOG)

Ο αλγόριθμος HOG είναι ένας περιγραφέας χαρακτηριστικών που παράγει ιστογράμματα υπολογίζοντας το πλάτος και την κατεύθυνση της κλίσης μιας τοπικής περιοχής στην εικόνα που εξετάζουμε. Ο αλγόριθμος αυτός είναι πολύ αποτελεσματικός και μπορεί στην εικόνα, λόγω της κατανομής των τοπικών κλίσεων να

χαρακτηρίσει το σχήμα και την τοπική εμφάνιση. Εξαιτίας της αποτελεσματικότητας και της ισχύς του, ο αλγόριθμος HOG χρησιμοποιείται ευρέως σε τομείς όπως η όραση υπολογιστών, στο πρόβλημα της αναγνώρισης πράξεων και μπορεί να παρέχει εξαιρετική απόδοση και σε σύνολα χαρακτηριστικών συμπεριλαμβανομένου των κυματισμών [39]. Η δομή του φαίνεται στο Σχήμα 2.7.

Ο αλγόριθμος Histogram of Oriented Gradients (HOG) έχει τα εξής πλεονεκτήματα:

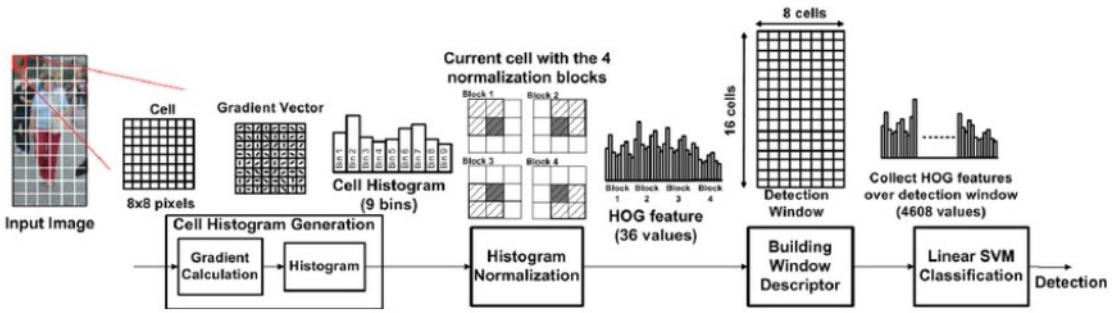
- Μπορεί να αναγνωρίζει αντικείμενα στην εικόνα ανεξάρτητα από το μέγεθος, τη θέση ή τον προσανατολισμό τους γιατί ο αλγόριθμος αυτός δεν αλλοιώνεται σε γεωμετρικές και φωτομετρικές μεταβολές.
- Το HOG μπορεί να λειτουργήσει χωρίς ισχυρή CPU, καθώς χρησιμοποιεί λιγότερη επεξεργαστική ισχύ για την αναγνώριση εικόνων μικρής κλίμακας.
- Η HOG χρησιμοποιείται σε καθημερινές εφαρμογές όπως στην ανίχνευση πεζών και στην ανάλυση ιατρικών εικόνων.

Και τα εξής μειονεκτήματα:

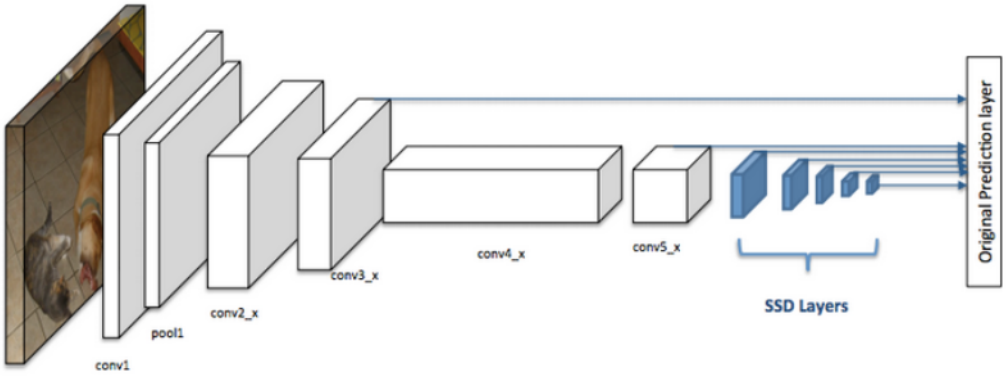
- Ο HOG έχει ευαισθησία στην περιστροφή εικόνας,δηλαδή δεν μπορεί να ανιχνεύσει με ακρίβεια ένα αντικείμενο εάν αυτό έχει περιστραφεί στην εικόνα [4].
- Το HOG χρησιμοποιεί μια τεχνική ολισθαίνοντος παραθύρου για την εξαγωγή χαρακτηριστικών από κάθε εικονοστοιχείο μιας εικόνας, η οποία έχει ως αποτέλεσμα αργό χρόνο επεξεργασίας κατά την αναγνώριση ενός αντικειμένου για εικόνες μεγάλης κλίμακας.Ως αποτέλεσμα, ενδέχεται να είναι λιγότερο ακριβής σε σχέση με τα σύγχρονα νευρωνικά δίκτυα συνελικτικής ανάλυσης.
- Ο HOG δεν είναι αποδοτικός και ακριβής σε μεγάλα datasets.

2.1.7 Single Shot Detector (SSD)

Ο αλγόριθμος Single Shot Detector (SSD) είναι ένα μοντέλο ανίχνευσης αντικειμένων σε πραγματικό χρόνο που με ένα μόνο πέρασμα σε μια εικόνα έχει τη δυνατότητα να προβλέπει την κατηγορία στην οποία ανήκει ένα αντικείμενο και να



Σχήμα 2.7: Παράδειγμα Histogram of Oriented Gradients

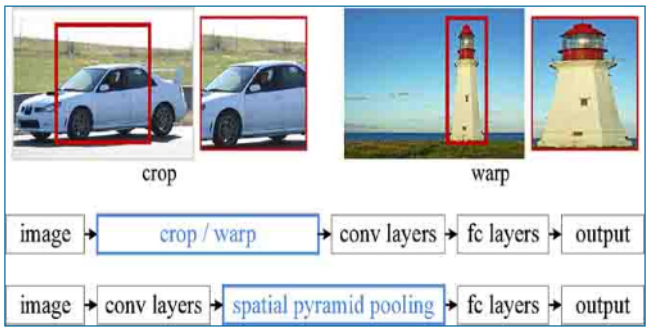


Σχήμα 2.8: Παράδειγμα Single Shot Detector
[6]

το τοποθετεί μέσα στον χώρο [6]. Ο αλγόριθμος SSD διαθέτει ένα βασικό νευρωνικό δίκτυο η αρχιτεκτονική του οποίου παρουσιάζεται στο (Σχήμα 2.8) και λειτουργεί εξάγοντας χαρακτηριστικά που ακολουθούνται από συνελικτικού τύπου πολλαπλά επίπεδα που κάνουν πρόβλεψη για την κατηγορία και τη θέση των αντικειμένων μέσα σε μια εικόνα [6]. Ο αλγόριθμος Single Shot Detector για να διαχειρίζεται πληθώρα μεγεθών στα φυσικά αντικείμενα συνδυάζει πολλαπλούς χάρτες χαρακτηριστικών για τη δημιουργία καλύτερων προβλέψεων [22]. Τέλος λειτουργεί με πλαίσια οριοθέτησης (bounding boxes) δηλαδή δημιουργεί k πλαίσια οριοθέτησης με διάφορα μεγέθη και αναλογίες διαστάσεων για κάθε σημείο της εικόνας που δίνεται. Κάθε πλαίσιο οριοθέτησης έχει τέσσερις μετατοπίσεις με βάση το προεπιλεγμένο σχήμα και c βαθμολογίες κλάσεων.

Ο αλγόριθμος έχει τα εξής πλεονεκτήματά:

- Ακόμη και με μειωμένο μέγεθος εικόνας εισόδου, ο αλγόριθμος SSD παρέχει σημαντικά υψηλότερη ακρίβεια σε σύγκριση με άλλες τεχνικές ενός σταδίου [22].



Σχήμα 2.9: Παράδειγμα Spatial Pyramid Pooling
[6]

- Εξαλείφει τη δημιουργία προτάσεων και τα επακόλουθα βήματα επαναδειγματοληψίας εικονοστοιχείων ή χαρακτηριστικών, σε αντίθεση με τα συμβατικά μοντέλα ανίχνευσης αντικειμένων που απαιτούν προτάσεις αντικειμένων, και ενοποιεί όλους τους υπολογισμούς σε ένα μόνο δίκτυο [22].

Και τα ακόλουθα μειονεκτήματα:

- Ο αλγόριθμος έχει χαμηλή ακρίβεια σε αντικείμενα μικρής κλίμακας [15].
- Στην περίπτωση που θέλουμε να αυξηθεί η ακρίβεια των αποτελεσμάτων που μας παρέχει ο αλγόριθμος η ταχύτητα μειώνεται καθώς χρειάζεται αύξηση του αριθμού των προκαθορισμένων πλαισίων οριοθέτησης [15].

2.1.8 Spatial Pyramid Pooling (SPP-net)

Ο αλγόριθμος Spatial Pyramid Pooling (SPP-net) αποτελεί μια τεχνική που επιτρέπει στα συνεπτυγμένα νευρωνικά δίκτυα (Concolution Neural Networks), να χειρίζονται εικόνες διαφόρων μεγεθών χωρίς να χάνουν χωρικές πληροφορίες [13]. Η αρχιτεκτονική του αλγορίθμου όπως φαίνεται και στο (Σχήμα 2.9) αποτελείται από ένα στρώμα SPP που το προσθέτει ο SPP-net που συγκεντρώνει τα χαρακτηριστικά και παράγει σταθερού μήκους εξόδους στην κορυφή του τελευταίου στρώματος συνελίξεων. Το στρώμα SPP πριν από τη διαβίβαση των αποτελεσμάτων στο επόμενο στρώμα, εκτελεί έναν αριθμό ξεχωριστών διαδικασιών συγκέντρωσης μεγέθους εξόδου και συνδυάζει τα αποτελέσματα. Ως αποτέλεσμα, το μοντέλο συνεπτυγμένου νευρωνικού δικτύου μπορεί να χρησιμοποιήσει εικόνα εισόδου οποιουδήποτε μεγέθους [13].

Ο αλγόριθμος Spatial Pyramid Pooling (SPP-net) έχει τα εξής πλεονεκτήματα:

- Διορθώνει τον περιορισμό για εικόνα εισόδου με σταθερό μέγεθος το οποίο επιτυγχάνεται γιατί ο SPP-net κάνει έναν συνδυασμό πληροφοριών μεταξύ των συνελικτικών στρωμάτων και των πλήρως συνδεδεμένων στρωμάτων σε ένα βαθύτερο επίπεδο της αρχιτεκτονικής του δικτύου [13].

- Είναι ακριβής και επιτυγχάνει κορυφαίες επιδόσεις καθώς επίσης μειώνει το υπολογιστικό κόστος της ανίχνευσης και ταξινόμησης αντικειμένων γιατί υπολογίζει τους χάρτες χαρακτηριστικών μόνο μια φορά από την εικόνα και εν συνεχεία δημιουργεί εξόδους σταθερού μήκους διαλέγοντας από αυθαίρετες περιοχές τα χαρακτηριστικά [13].

Και τα εξής μειονεκτήματα:

- Επειδή στο στρώμα SPP πραγματοποιούνται πολλές πράξεις ο αλγόριθμος αυτός μπορεί να είναι υπολογιστικά ακριβός [14].

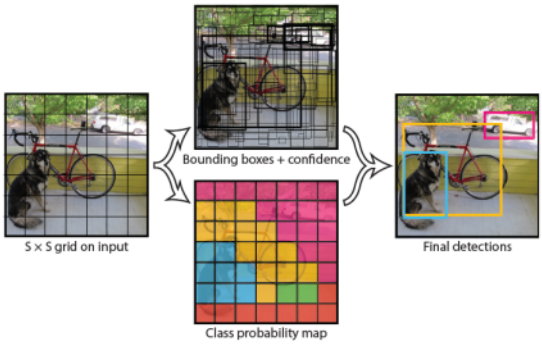
- Παρουσιάζει μια έλλειψη αποτελεσματικότητας στον χειρισμό εικόνων με μεγάλες διακυμάνσεις στις αναλογίες διαστάσεων και στην κλίμακα [14].

2.1.9 You Only Look Once (YOLO)

Ο αλγόριθμος YOLO, που προτάθηκε από τον Joseph Redmond από το Πανεπιστήμιο της Ουάσιγκτον, είναι ένας αλγόριθμος αναγνώρισης αντικειμένων σε πραγματικό χρόνο [29]. Το συμπαγές μέγεθος και η γρήγορη ταχύτητα υπολογισμού του μοντέλου αποτελούν τη βάση της τεχνικής αναγνώρισης στόχου YOLO. Η οργάνωση του YOLO είναι απλή. Μέσω του νευρωνικού δικτύου, μπορεί να εξάγει αμέσως τη θέση και την κατηγορία του οριοθετημένου πλαισίου. Επειδή ο YOLO χρειάζεται απλώς να φορτώσει την εικόνα στο δίκτυο για να λάβει το τελικό αποτέλεσμα ανίχνευσης, η ταχύτητά του είναι γρήγορη. Ο αλγόριθμος αυτός μειώνει το σφάλμα ανίχνευσης του φόντου ως αντικειμένου διότι μπορεί και κωδικοποιεί τις παγκόσμιες πληροφορίες κάνοντας χρήση για ανίχνευση την παγκόσμια εικόνα [20]. Η δομή του YOLO φαίνεται στο παρακάτω Σχήμα 2.10.

Ο αλγόριθμος YOLO έχει τα εξής πλεονεκτήματα:

- Έχει ισχυρή ικανότητα γενίκευσης επειδή ο YOLO μπορεί να μάθει ιδιαίτερα γενικευμένα χαρακτηριστικά για να μεταφερθούν σε άλλα πεδία [20].



Σχήμα 2.10: Παράδειγμα YOLO [17]

- Μπορεί να κάνει ανίχνευση αντικειμένων σε πραγματικό χρόνο[20].
- Είναι πολύ γρήγορος αλγόριθμος[20].

Και τα εξής μειονεκτήματα:

- Για αντικείμενα που βρίσκονται πολύ κοντά το ένα στο άλλο τα αποτελέσματα των δοκιμών του YOLO είναι φτωχά καθώς παρουσιάζει χαμηλότερη ανάκληση και σφάλμα εντοπισμού[20].
- Δεν έχει τα επιθυμητά αποτελέσματα σε αντικείμενα που έχουν μικρό μέγεθος [20].

2.1.10 Σύγκριση

Ανάμεσα στους αλγορίθμους R-CNN, Fast R-CNN και Faster R-CNN καταλήγουμε ότι υπάρχουν διαφοροποιήσεις με βάση την απόδοσή τους και συμπεραίνουμε ότι ο Faster R-CNN είναι αποδοτικότερος από το Fast R-CNN που με τη σειρά του είναι πιο αποδοτικός από το R-CNN. Το συμπέρασμά αυτό επιβεβαιώνεται από τις παρακάτω συγκρίσεις:

- Ο αλγόριθμος Fast R-CNN μοιράζει τους υπολογισμούς σε πολλαπλές προτάσεις σε αντίθεση με τον αλγόριθμο R-CNN στον οποίο κάθε πρόταση περιοχής τροφοδοτείται ανεξάρτητα από τις υπόλοιπες, το οποίο στην πράξη έχει χρονικό κόστος διότι χρειάζονται S*N δευτερόλεπτα. Όπου S συμβολίζονται τα δευτερόλεπτα που απαιτούνται για την επεξεργασία μιας περιοχής και το N χρησιμοποιείται για τον συμβολισμό του συνόλου των περιοχών. Οπότε γίνεται κατανοητό ότι ο Fast R-CNN είναι ταχύτερος από τον αλγόριθμο R-CNN.

- Ο αλγόριθμος Faster R-CNN σε μεγάλης κλίμακας σύνολο δεδομένων προσώπου εξάγει αποτελέσματα πιο ακριβή και με μεγαλύτερη ταχύτητα επεξεργασίας σε σχέση με τον αλγόριθμο Fast R-CNN [19]

Ο αλγόριθμος R-FCN είναι πιο αποδοτικός από τους αλγόριθμους Fast R-CNN και Faster R-CNN διότι δεν εφαρμόζει ένα δαπανηρό υποδίκτυο ανά περιοχή εκατοντάδες φορές αφού ο αλγόριθμος R-FCN είναι πλήρως συνελικτικός και σχεδόν όλοι οι υπολογισμοί μοιράζονται σε ολόκληρη την εικόνα [5]. Επιπλέον χρησιμοποιεί ένα ενιαίο δίκτυο για την πρόβλεψη των κατηγοριών αντικειμένων και τη λεπτομερή ρύθμιση των θέσεων των αντικειμένων. Σε αντίθεση με τον αλγόριθμο R-FCN ο αλγόριθμος HOG είναι ένας περιγραφέας χαρακτηριστικών που μετρά περιπτώσεις προσανατολισμού κλίσης σε συγκεκριμένες περιοχές μιας εικόνας [40]. Η βασική διαφορά ανάμεσα στον αλγόριθμο SPF-net και στον αλγόριθμο SSD είναι ότι ο πρώτος μπορεί να χαρακτηριστεί ως ανιχνευτής ενός σταδίου που εξάγει χαρακτηριστικά πολλαπλών κλιμάκων χρησιμοποιώντας μια χωρική πυραμίδα, ενώ ο δεύτερος χαρακτηρίζεται ως ανιχνευτής ενός σταδίου που αυξάνει το καρέ σε μια εικόνα ανά δευτερόλεπτο κατά περίπου πέντε φορές σε σχέση με τον Faster R-CNN. Τέλος ο αλγόριθμος YOLO έχει τα εξής χαρακτηριστικά που τον καθιστούν έναν πολύ αξιόλογο αλγόριθμο, έχει ισχυρή ικανότητα γενίκευσης επειδή ο YOLO μπορεί να μάθει ιδιαίτερα γενικευμένα χαρακτηριστικά για να μεταφερθούν σε άλλα πεδία, μπορεί να κάνει ανίχνευση αντικειμένων σε πραγματικό χρόνο και τέλος είναι πολύ γρήγορος αλγόριθμος [20]. Το συμπέρασμα που προκύπτει είναι ότι η επιλογή αλγορίθμου που μας εξυπηρετεί μπορεί να διαφέρει από περίπτωση σε περίπτωση καθώς εξαρτάται από το μέγεθος του αντικειμένου, τη θέση του στον χώρο, την ταχύτητα και την ακρίβεια που επιθυμούμε.

2.2 Αλγόριθμοι Κρυπτογράφησης

Οι αλγόριθμοι κρυπτογραφίας χρησιμοποιούνται για την προστασία της επικοινωνίας από ξένους. Υπάρχουν πολλά διαφορετικά είδη κρυπτογράφησης, όπως η κρυπτογράφηση που βασίζεται σε νευρωνικά δίκτυα και οι τεχνικές κρυπτογράφησης εικόνων. Οι εικόνες κρυπτογραφούνται με τη χρήση μεθόδων κρυπτογράφησης εικόνων, οι οποίες επίσης αυξάνουν την ασφάλεια της επικοινωνίας [7]. Οι αλγόριθ-

μοι κρυπτογράφησης χωρίζονται σε δύο κατηγορίες που είναι η συμμετρική και η ασύμμετρη κρυπτογράφηση.

- Συμμετρική κρυπτογράφηση: Οι αλγόριθμοι που χρησιμοποιούνται για αυτού του είδους την κρυπτογράφηση έχουν συμμετρικό κλειδί χρησιμοποιούν δηλαδή το ίδιο κλειδί τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση.

- Ασύμμετρη Κρυπτογράφηση: Οι αλγόριθμοι που χρησιμοποιούνται για αυτού του είδους την κρυπτογράφηση έχουν ασύμμετρο κλειδί χρησιμοποιούν δηλαδή διαφορετικά κλειδιά για την κρυπτογράφηση και την αποκρυπτογράφηση.

Υπάρχουν και άλλοι αλγόριθμοι κρυπτογράφησης όπως είναι η κρυπτογράφηση με χρήση συνάρτησης κατακερματισμού.

2.2.1 Συμμετρική Κρυπτογράφηση

DES

Ο αλγόριθμος κρυπτογράφησης DES (Data Encyption Standard) βασίστηκε στο κρυπτογράφημα Lucifer για τη δημιουργία του και έγινε πρότυπο το 1974. Ο DES μετατρέπει ένα μπλοκ εισόδου 64-bit σε ένα μπλοκ εξόδου 64-bit χρησιμοποιώντας ένα κλειδί 56-bit. Στο κλειδί αυτό προστίθενται άλλα 8-bit περιττής ισοτιμίας με αποτέλεσμα το τελικό μέγεθος του κλειδιού να είναι 64-bit. Ο αλγόριθμος αυτός όμως ήταν ένα αδύναμο κρυπτογραφικό μπλοκ που αποτελούσε στόχο πολλών επιθέσεων. Για αυτόν τον λόγο δημιουργήθηκε μια βελτιωμένη έκδοση του, ο 3DES (Triple DES) που χρησιμοποιούσε τρία κλειδιά σε σχέση με το ένα που χρησιμοποιούσε ο DES με αποτέλεσμα να είναι πιο ασφαλής [35].

AES

Ο AES (Advanced Encryption Standard) είναι ένας συμμετρικός αλγόριθμος δηλαδή χρησιμοποιεί το ίδιο κλειδί τόσο για την κρυπτογράφηση όσο και για την αποκρυπτογράφηση του μηνύματος. Κρυπτογραφεί δεδομένα με μορφή μπλοκ που έχουν μέγεθος 128bits. Το μέγεθος των κλειδιών που χρησιμοποιεί για την αποκρυπτογράφηση μπορεί να είναι 128 bit είτε 192 bit είτε 256 bit. Δημιουργήθηκε γα

να αντικαταστήσει τον αλγόριθμο DES και για αυτό χρησιμοποιείται παγκοσμίος ως καλύτερη λύση από αυτόν και τον 3DES. Αυτή η διαφορά στην απόδοση οφείλεται στο γεγονός ότι ο AES δεν χρησιμοποιεί ένα δίκτυο Fiestel, καθώς αποτελεί μια παραλλαγή του Rijindael που χαρακτηρίζεται από σταθερό μέγεθος μπλοκ που είναι 128 bit και μέγεθος κλειδιού 128bit ή 192bit ή 256bit. Στην πραγματικότητα ο Rijindael καθορίζεται με μεγέθη μπλοκ και κλειδιών που μπορούν να είναι οποιαδήποτε πολλαπλάσιο των 32 bits, και τα δύο έχουν εύρος 128-256 bits [35, 30]. Ο σχεδιασμός του αλγορίθμου AES βασίζεται στο δίκτυο αντικατάστασης-μεταλλαγής και είναι γρήγορος τόσο σε λογισμικό όσο και σε υλικό. Ο αλγόριθμος αυτός λειτουργεί με bytes που είναι σε μορφή πίνακα 4χ4 μείζονος τάξης, βέβαια μερικές εκδόσεις του Rijindael έχουν μεγαλύτερο μέγεθος μπλοκ και διαθέτουν επιπλέον στήλες, η πλειοψηφία των υπολογισμών που πραγματοποιεί ο AES γίνονται σε ένα πεπερασμένο πεδίο [30].

Blowfish

Ο blowfish είναι ένας αλγόριθμος συμμετρικής κρυπτογράφησης που χρησιμοποιεί δεδομένα που βρίσκονται σε μορφή μπλόκ. Χρειάζεται ένα κλειδί με μεταβλητό μήκος το οποίο κυμαίνεται από 32bit-448bit για να πραγματοποιήσει την κρυπτογράφηση και την αποκρυπτογράφηση. Σχεδιάστηκε το 1993 από τον Bruce Schneier [32] , ως μια εναλλακτική των αλγορίθμων κρυπτογράφησης που ήδη υπήρχαν απλά αποτελούσε μια γρήγορη και δωρεάν λύση. Χρησιμοποιείται για την αποτελεσματική κρυπτογράφηση και διασφάλιση των δεδομένων, γιατί παρόλο που παρουσιάζει αδυναμία στα κλειδιά που χρησιμοποιεί δεν έχει γίνει κάποια επιτυχημένη επίθεση για την κατάρριψη της κρυπτογράφησης που προσφέρει [35].

2.2.2 Ασύμμετρη Κρυπτογράφηση

RSA

Ο RSA είναι ένας αλγόριθμος ασύμμετρης κρυπτογράφησης δηλαδή χρησιμοποιεί δύο διαφορετικά κλειδιά το δημόσιο κλειδί για κρυπτογράφηση και το ιδιωτικό κλειδί για αποκρυπτογράφηση.Λόγω της ύπαρξης των δυο κλειδιών μόνο τα άτομα που έχουν τα σωστά κλειδιά μπορούν να πραγματοποιούν τη διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης ενός μηνύματος. Τα δύο κλειδιά

δεν πρέπει να έχουν σχέση μεταξύ τους έτσι ώστε η αποκρυπτογράφηση να μην επιτυγχάνεται από τρίτους που δεν πρέπει να γνωρίζουν το περιεχόμενο του μηνύματος.Η κρυπτογράφηση προκύπτει μέσα από μια διαδικασία παραγοντοποιήσης μεγάλων πρώτων αριθμών και για αυτόν τον λόγο μέχρι στιγμής δεν έχει επικυρωθεί κάποια γνωστή απόπειρα να τον σπάσουν. Ο αλγόριθμος λειτουργεί ως εξής, έχουμε το μήνυμα το οποίο το κρυπτογραφούμε με το δημόσιο κλειδί και στη συνέχεια το αποστέλλουμε σε έναν άλλο χρήστη μέσα από έναν δίαυλο επικοινωνίας. Μετά ο δέκτης αποκρυπτογραφεί το μήνυμα με το ιδιωτικό κλειδί έτσι ώστε να επιτευχθεί η επικοινωνία μεταξύ των δυο πλευρών. Το κρυπτογραφημένο και το αποκρυπογραφημένο μήνυμα υπολογίζονται μέσα από μαθηματικούς τύπους[25].

DSS

Ο DSS (Digital Signature Standard) είναι ένας αλγόριθμος ασύμμετρης κρυπτογράφησης, που εφαρμόζει έναν ειδικά προσαρμοσμένο αλγόριθμο που αφορά τις ψηφιακές υπογραφές και δεν διαθέτει τη δυνατότητα ανταλλαγής κλειδιών και κρυπτογράφησης που έχει ο RSA. Ο αλγόριθμος αυτός είναι μια τεχνική δημόσιου κλειδιού, η διαδικασία επαλήθευσης της υπογραφής περιλαμβάνει τη χρήση το δημόσιο κλειδί για την αποκωδικοποίηση της υπογραφής και στη συνέχεια τη σύγκριση των αποτελεσμάτων με το αρχικό κατακερματισμένο μήνυμα. που υπολογίζεται με τη χρήση του μυστικού κλειδιού. Ο αλγόριθμος λειτουργεί ως εξής χρησιμοποιεί δυο πολύ μεγάλους ακέραιους αριθμούς που ο καθένας αναπαριστάται στον υπολογιστή ως συμβολοσειρά δυαδικών ψηφίων, σε αυτή την περίπτωση ο αλγόριθμος χρησιμοποιείται τόσο για τη δημιουργία όσο και για την επικύρωση των υπογραφών που δημιουργούνται με τη χρήση του ιδιωτικού κλειδιού και μπορούν να επαληθευτούν με το δημόσιο κλειδί που ταιριάζει το ιδιωτικό αλλά διαφέρει από αυτό [33].

ElGamal

Ο ElGamal είναι αλγόριθμος ασύμμετρης κρυπτογράφησης δηλαδή χρησιμοποιεί για τη διαδικασία της κρυπτογράφησης και της αποκρυπτογράφησης τόσο το δημόσιο όσο και το ιδιωτικό κλειδί. Αποτελεί μια εναλλακτική μέθοδο για την κρυπτογράφηση δημόσιου κλειδιού RSA αλλά έχουν μια βασική διαφορά μεταξύ τους και αυτή συναντάτε στο γεγονός ότι η ασφάλεια του RSA βασίζεται στη μαθηματική δυ-

σκολία παραγοντοποίησης μεγάλων πρώτων αριθμών ενώ αντίθετα η ασφάλεια του ElGamal βασίζεται στη δυσκολία υπολογισμού του διακριτού λογαρίθμου modulus μεγάλων πρώτων αριθμών [16, 38]. Χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση σημάτων ομιλίας και η διαδικασία είναι η εξής το σήμα αρχικά κρυπτογραφείται από τον αποστολέα με χρήση του αλγορίθμου ElGamal και αποστέλλεται στον παραλήπτη μέσω ενός διαύλου επικοινωνίας. Στη συνέχεια ο παραλήπτης λαμβάνει το μήνυμα και το αποκρυπτογραφεί με αποτέλεσμα να έχει ως τελικό προϊόν το αρχικό μήνυμα που στάλθηκε εφόσον δεν έχουν επέμβει εξωτερικοί παράγοντες [16]. Είναι πολύ ισχυρός αλγόριθμος γιατί βασίζεται στον διακριτό λογαρισμό που είναι ένα ιδιαίτερα δύσκολο πρόβλημα στα μαθηματικά, επειδή εξαρτάται κυρίως από τη σύζευξη για να προκύψουν όλες οι πιθανές λύσεις για αυτό, οπότε απαιτείται μεγάλο χρονικό διάστημα για να σπάσει οπότε θεωρείται σχεδόν ανέφικτο. Χρησιμοποιώντας τον ElGamal το ίδιο μήνυμα απλού κειμένου οδηγεί σε διαφορετικό μήνυμα-κρυπτογράφημα κάθε φορά που κρυπτογραφείται [38].

2.2.3 Σύγκριση

Κάνοντας μια σύγκριση για όλους τους αλγορίθμους καταλήγουμε ότι ο Blowfish είναι ταχύτερη από τον αλγόριθμο DES και τον αλγόριθμο AES στις περισσότερες περιπτώσεις καθώς υπάρχουν συγκεκριμένες περιπτώσεις [18] που τα αποτελέσματα έδειξαν ότι η απόδοση του AES είναι καλύτερη από αυτή του Blowfish. Στην περίπτωση που θέλουμε να κρυπτογραφήσουμε μια εικόνα και έχουμε επιλέξει να την επεξεργαστούμε ως τύπο δεδομένων αντί για αρχείο κειμένου έχει μελετηθεί ότι ο αλγόριθμος Blowfish καταναλώνει περισσότερο χρόνο σε σχέσει με τους αλγορίθμους AES, DES και 3DES, με τον DES να παραμένει ταχύτερος σε σχέση με τον 3DES. Για ένα μικρού μεγέθους αρχείο το οποίο βρίσκεται στο νέφος παρατηρείται ότι έχει την υψηλότερη ταχύτητα ενώ ο αλγόριθμος RSA είναι ο πιο χρονοβόρος. Γενικότερα ισχύει ότι ο χρόνος κρυπτογράφησης αυξάνεται όσο αυξάνεται το μέγεθος των δεδομένων δηλαδή όσο περισσότερα bytes είναι το αρχείο τόσο περισσότερος χρόνος χρειάζεται.Και επιπλέον ο χρόνος αυτός δεν επηρεάζεται από τον τύπο των δεδομένων ενός αρχείου.Όσο αφορά τους αλγόριθμους RSA και ElGamal, ύστερα από πειράματα και μελέτες προέκυψαν τα εξής συμπεράσματα

ο χρόνος κρυπτογράφησης του RSA είναι καλύτερος από τον ElGamal αλλά ο χρόνος αποκρυπτογράφησης του ElGamal είναι καλύτερος σε σύγκριση με τον RSA, η απόδοση της διαδικασίας κρυπτογράφησης RSA είναι καλύτερη και η απόδοση στη διαδικασία αποκρυπτογράφησης του ElGamal είναι καλύτερη από την απόδοση του RSA οπότε μπορεί να υπάρχει το συμπέρασμα ότι ο αλγόριθμος RSA έχει καλύτερη απόδοση από τον αλγόριθμο ElGamal [24]. Ο αλγόριθμος DSS είναι ταχύτερο στην υπογραφή αλλά πιο αργό στην επαλήθευση. Ο RSA παράγει κλειδιά πιο αργά αλλά η κρυπτογράφηση που προσφέρει είναι πιο γρήγορη σε σχέση με τον DSS και για την ανταλλαγή κλειδιών απαιτεί λιγότερα bytes σε σχέση με τον ElGamal. Το συμπέρασμα που προκύπτει σχετικά με το ποιος αλγόριθμος είναι πιο αποδοτικός είναι ότι αυτό εξαρτάται από τα δεδομένα κάθε περίπτωσης δηλαδή το απαιτούμενο επίπεδο ασφάλειας, τον όγκος των δεδομένων και τους διαθέσιμους πόρους του υπολογιστικού μας συστήματος.

Κεφάλαιο 3

Βιβλιογραφική ανασκόπηση

Το αντικείμενο της συγκεκριμένης διπλωματικής έχει ερευνηθεί και μελετηθεί και από άλλους επιστήμονες μέσα σε βάθος χρόνου, οι οποίοι έκαναν παρόμοιες ή και πολύ διαφορετικές επιλογές τόσο στο αντικείμενο το οποίο μελέτησαν δηλαδή στη φύση των εικόνων που επέλεξαν, όσο και στην επιλογή αλγορίθμων τόσο για την αναγνώρισή τους όσο και για την κρυπτογράφηση.

3.1 Η Φύση των εικόνων που χρησιμοποιήθηκαν

Η αναγνώριση αντικειμένων και η κρυπτογράφηση σε εικόνες έχει πολλές εφαρμογές και για αυτόν τον λόγο έχουν χρησιμοποιηθεί διαφορετικά είδη εικόνων. Ένα από αυτά είναι οι εικόνες ασθενειών σε φυτά. Η διαδικασία της αναγνώρισης είναι η εξής, αρχικά εντοπίζεται η περιοχή στην οποία εμφανίζεται η ασθένεια και στη συνέχεια μέσα από επεξεργασία είτε εκείνη τη στιγμή είτε σε κάποιο εργαστήριο με τη χρήση κατάλληλων αλγορίθμων γίνεται ο προσδιορισμός και η εξαγωγή του αποτελέσματος [8]. Σε κάποιες άλλες μελέτες έχουν χρησιμοποιηθεί αρχεία εικόνας, ήχου και βίντεο από κάμερες ασφαλείας, στα οποία γίνεται αναγνώριση αντικειμένων από τους κατάλληλους αλγόριθμους και στη συνεχεία εφαρμόζεται κρυπτογραφία στις εικόνες ή στα αντικείμενα που περιέχονται σε αυτές έτσι ώστε να μην είναι ανιχνεύσιμα, και να μην καθίσταται εύκολη η επεξεργασία και η συμπίεση της εικόνας [3, 1]. Ένα άλλο είδος εικόνων που χρησιμοποιούνται για τον σκοπό της αναγνώρισης αλλά και κρυπτογράφησης εικόνων είναι ιατρικές εικόνες. Στον τομέα αυτό η αναγνώριση των αντικειμένων είναι χρήσιμη διότι ο υπολογισμός και ο προσδιορισμός του τι απεικονίζεται στην εικόνα γίνεται πιο γρήγορα και με μεγαλύτερη ακρίβεια. Η κρυπτογράφηση αυτών των εικόνων συμβάλει στην

εξασφάλιση ότι τα ευαίσθητα αυτά δεδομένα δεν θα κοινοποιηθούν σε τρίτους που δεν έχουν την απαραίτητα άδεια [23].

3.2 Ποιοι αλγόριθμοι αναγνώρισης αντικειμένων χρησιμοποιήθηκαν

Για την αναγνώριση αντικειμένων σε εικόνες έχουν χρησιμοποιηθεί διαφορετικοί αλγόριθμοι. Οι πιο συχνά χρησιμοποιούμενοι που ξεχωρίζουν για την απόδοση, την ακρίβεια και τον χρόνο που εξάγουν αποτελέσματα είναι ο SSD, ο Faster R-CNN και ο YOLO. Ο αλγόριθμος Faster R-CNN αντικατέστησε αλγορίθμους όπως είναι ο αλγόριθμος R-CNN ο οποίος χρησιμοποιεί το μοντέλο ολισθαίνοντος παραθύρου, ενώ ο Faster R-CNN χρησιμοποιεί μια ξεχωριστή εκπαίδευση με βαθύ δίκτυο συνέλιξης για την απομόνωση χαρακτηριστικών και επιπλέον χρησιμοποιεί μηχανές διανυσμάτων υποστήριξης για κατηγοριοποίηση των αντικειμένων μιας εικόνας. Αρχικά ο αλγόριθμος R-CNN αντικαταστάθηκε από τον αλγόριθμο Fast R-CNN οποίος είναι εννιά φορές ταχύτερος στην εκπαίδευση και συνδυάζει την εξαγωγή χαρακτηριστικών με την ταξινόμηση σε ένα πλαίσιο ταξινόμησης. Στη συνέχεια αντικαταστάθηκε από τον Faster R-CNN καθώς τοποθετεί την περιοχή απομόνωσης σε ένα πρότυπο δικτύου που αναφέρεται ως δίκτυο πρότασης περιοχής (RPN), παρόλο που η ακρίβεια ανάμεσα στους δύο αλγορίθμους παραμένει η ίδια. Ο αλγόριθμος YOLO επιτυγχάνει την ανίχνευση στόχου από άκρη σε άκρη λαμβάνοντας από την αρχική εικόνα την κατηγορία και τη θέση και βγάζοντας την κατάλληλη έξοδο παρέχοντας ακρίβεια. Τέλος ο SSD χρησιμοποιεί ένα μόνο βαθύ νευρωνικό δίκτυο και εξαλείφει τα pixel και την επαναδειγματοληψία καθώς απαιτεί μια μόνο πρόταση αντικειμένου γιατί βασίζεται στην πλήρη εξάλειψη της διαδικασίας που παράγει μια πρόταση. Το θετικό χαρακτηριστικό του SSD και για αυτό τον λόγο προτιμάται, είναι ότι χρησιμοποιεί πολυκλιμακωτές εξόδους συνελικτικού πλαισίου οριοθέτησης που συνδέονται με διάφορους χάρτες χαρακτηριστικών οπότε η εκπαίδευση είναι πολύ εύκολη όταν πρόκειται να ενσωματωθεί σε ένα σύστημα [34].

3.3 Ποιοι αλγόριθμοι κρυπτογράφησης χρησιμοποιήθηκαν

Για την κρυπτογράφηση εικόνων χρησιμοποιούνται ποικίλοι αλγόριθμοι κρυπτογράφησης που διαφέρουν μεταξύ τους ανάλογα με τον τρόπο που πραγματοποιούν την κρυπτογράφηση. Οι αλγόριθμοι αυτοί είναι ο AES, ο Blowfish, ο αλγόριθμος κρυπτογράφησης εικόνων με χρήση αυτοαναιρέσιμου πίνακα κλειδιών του Hill Cipher, ο αλγόριθμος κρυπτογράφησης εικόνων που συνδυάζει τη τεχνική μεταβολής ακολουθούμενη από κρυπτογράφηση, ο αλγόριθμος κρυπτογράφησης εικόνων με χρήση της προηγμένης κρυπτογράφησης Hill, ο DES, ο SHA-512, ο αλγόριθμος κρυπτογράφησης εικόνων που βασίζεται σε σύνθεση δύο χαοτικών λογιστικών χαρτών, ο αλγόριθμος κρυπτογράφησης εικόνων που χρησιμοποιεί μετασχηματισμούς AFFINE και XOR, η ασφάλεια εικόνων μέσω γενετικών αλγορίθμων και τέλος ο αλγόριθμος κρυπτογράφησης εικόνων που βασίζεται στη γενική προσέγγιση για πολλαπλά χαοτικά συστήματα [27]. Ένας άλλος αλγόριθμος που έχει χρησιμοποιηθεί με σκοπό την κρυπτογράφηση εικόνων είναι ο RSA που ανήκει στην κατηγορία αλγορίθμων ασύμμετρης κρυπτογράφησης και επιλέγεται από τους ερευνητές καθώς κάνοντας χρήση αυτού επιτυγχάνεται η ασφάλεια και η αυθεντικοποίηση των πληροφοριών καθώς επίσης επιλύει το σημαντικό πρόβλημα στην παραγοντοποίηση μεγάλων ακέραιων αριθμών[31]. Τέλος για την κρυπτογράφηση εικόνων έχουν χρησιμοποιηθεί για μελέτες αλγόριθμοι όπως κρυπτογράφηση με βάση χαοτικό χάρτη, κρυπτογράφηση με βάση την κρυπτογραφία ελλειπτικής καμπύλης (ECC),κρυπτογράφηση με βάση μεθόδους DNA, κρυπτογράφηση με βάση διάφορα σχήματα μαθηματικών μοντέλων [28].

3.4 Σύγκριση

Στον Πίνακα 3.1 παρουσιάζονται συνοπτικά ποιοι αλγόριθμοι, που αναφέρθηκαν πιο πάνω, χρησιμοποιήθηκαν σε άλλες μελέτες:

Πίνακας 3.1: Αλγόριθμοι και έγγραφα στα οποία αναφέρονται.

Αλγόριθμος αναγνώρισης αντικειμένων	**[8]**	**[34]**	**[26]**	**[2]**
RCNN	x	x	x	x
Fast RCNN	-	x	-	x
Faster RCNN	x	x	x	x
R-FCN	x	-	-	-
HOG	x	-	-	-
SSD	x	x	-	-
SPP-net	-	-	-	-
YOLO	-	x	x	x
Αλγόριθμος κρυπτογράφησης	**[27]**	**[31]**	**[37]**	**[28]**
DES	x	-	-	-
AES	x	x	x	x
Blowfish	x	-	-	-
RSA	-	x	-	-
DSS	-	-	-	-
El Gamal	-	-	-	-

Κεφάλαιο 4

Υλοποίηση

Τα πειράματα έγιναν σε έναν hp workstation Z6 G4 με 2 επεξεργαστές Intel Xeon Silver 4114@2,2 GHz, 8GB μνήμης και δίσκο NVme 256GB και μια κάρτα γραφικών NVIDIA Quadro P620. Σε αυτό το τερματικό εγκαταστάθηκε λειτουργικό σύστημα Linux Mint 64x (kernel 5.15.0). Επιλέχθηκε η χρήση της εφαρμογής PyCharm Community Edition 2023.2 για τη συγγραφή και εκτέλεση του κώδικα Python που απαιτήθηκε με σκοπό την υλοποίηση αυτής της εργασίας.

Έγινε η απαραίτητη έρευνα στο διαδίκτυο και εξετάστηκαν σε θεωρητικό επίπεδο τα papers των αλγορίθμων αναγνώρισης αντικειμένων και κρυπτογράφησης. Από την έρευνα αυτή προέκυψε ότι:

- Όσον αφορά στους αλγορίθμους αναγνώρισης αντικειμένων που περιγράφονται παραπάνω, εξετάστηκαν οι αλγόριθμοι Faster RCNN και YOLO που είναι οι πιο διαδεδομένοι ακριβείς και πιο αποτελεσματικοί για τη συγκεκριμένη εργασία. Ο αλγόριθμος HOG δεν προτείνεται για αναγνώριση αντικειμένων σε φωτογραφίες γιατί η αποτελεσματικότητά του επηρεάζεται σημαντικά από τη γωνία λήψης απεικόνισης του αντικειμένου σε κάθε φωτογραφία. Ο αλγόριθμος SSD παρουσιάζει πολλά κοινά χαρακτηριστικά και σε πολλά σημεία η λειτουργία του ταυτίζεται με τον YOLO αφού και οι δύο είναι αλγόριθμοι αναγνώρισης αντικειμένων σε πραγματικό χρόνο.

- Από τους αλγόριθμους κρυπτογράφησης και αποκρυπτογράφησης που αναλύθηκαν, έχουν επιλεγεί να υλοποιηθούν οι τρεις αλγόριθμοι συμμετρική κρυπτογράφησης, DES, AES, BlowFish, καθώς οι αλγόριθμοι ασύμμετρης κρυπτογράφησης DSS και ElGamal χρησιμοποιούνται για τη ψηφιακή υπογραφή

κατά κύριο λόγο, ενώ ο αλγόριθμος RSA δεν μπορεί να πραγματοποιήσει κρυπτογράφηση εικόνων και για αυτό τον σκοπό χρησιμοποιείται σε συνδυασμό με κάποιον αλγόριθμο συμμετρικής κρυπτογράφησης συνήθως τον αλγόριθμο AES, οπότε η κρυπτογράφηση γίνεται με συμμετρικό τρόπο.

4.1 Ο αλγόριθμος Faster RCNN

4.1.1 Γενικά

Ο αλγόριθμος Faster RCNN συγκεντρώνει χαρακτηριστικά από τους αλγορίθμους RCNN, Fast RCNN και R-FCNN δηλαδή από αυτούς που ανήκουν στη ευρύτερη κατηγορία των αλγορίθμων αναγνώρισης αντικειμένων συνελικτικού νευρωνικού δικτύου και είναι πιο γρήγορος χρονικά και πιο ακριβής στα αποτελέσματά του σε σχέση με τους προαναφερθέντες. Ο αλγόριθμος Faster-RCNN χρησιμοποιείται για ανίχνευση αντικειμένων σε εικόνες σε αρκετές περιπτώσεις. Η διαδικασία με την οποία λειτουργεί είναι η εξής: αρχικά χρησιμοποιεί ένα ConvNet με σκοπό την εξαγωγή χαρτών χαρακτηριστικών από τις δοθείσες εικόνες, στη συνέχεια αυτοί οι χάρτες εισάγονται σε ένα δίκτυο προτάσεων περιοχής που εξάγει σαν αποτέλεσμα τα υποψήφια πλαίσια οριοθέτησης. Εν συνεχεία επειδή τα υποψήφια πλαίσια οριοθέτησης πρέπει να είναι στο ίδιο μέγεθος εφαρμόζουμε ένα επίπεδο συγκέντρωσης ROI σε αυτά. Τέλος οι προτάσεις περιοχών περνούν σε ένα πλήρως συνδεδεμένο επίπεδο έτσι ώστε να ταξινομηθούν και να οριοθετηθούν τα πλαίσια για τα αντικείμενα που θέλουμε να αναγνωρίσουμε. Ο αλγόριθμος αυτός προϋποθέτει την ύπαρξη ενός συνόλου δεδομένων αρκετά μεγάλο που χρησιμοποιούνται για την εκπαίδευση του αλγορίθμου έτσι ώστε να είναι δυνατή η αναγνώριση μοτίβων και χαρακτηριστικών. Τις περισσότερες φορές τα σύνολα δεδομένων περιέχουν πολλαπλές κατηγορίες του ίδιου αντικειμένου για την καλύτερη και αποτελεσματικότερη εκπαίδευση του αλγορίθμου Faster R-CNN.

4.1.2 Τρόπος Υλοποίησης αλγορίθμου Faster R-CNN

Για την υλοποίηση της αναγνώρισης αντικειμένων είναι απαραίτητο να γίνουν τα εξής βήματα:

- Συλλογή δεδομένων δηλαδή απόκτηση του κατάλληλου συνόλου δεδομένων

που περιέχουν τόσο το αντικείμενο που μας ενδιαφέρει, όσο και άλλα αντικείμενα μη συναφή.

- Σχολιασμός δεδομένων επισημαίνοντας στις φωτογραφίες τις περιοχές ενδιαφέροντος (ROI) δηλαδή τις περιοχές στις οποίες εντοπίζεται το αντικείμενο ενδιαφέροντος και χρησιμοποιούνται ως ετικέτες βασικής αλήθειας για την κατάλληλη εκπαίδευση του μοντέλου Faster R-CNN.

- Προεπεξεργασία δεδομένων που περιλαμβάνει την αλλαγή μεγέθους, την κανονικοποίηση και την επαύξηση των εικόνων και των σχολιασμών που δημιουργήθηκαν από το προηγούμενο βήμα.

- Επιλογή κατάλληλου μοντέλου που συνήθως είναι προ-εκπαιδευμένο και έχει εκπαιδευτεί σε σύνολα δεδομένων εικόνας μεγάλης κλίμακας όπως είναι το resnet50.

- Εκπαίδευση του μοντέλου διαχωρίζοντας τα δεδομένα σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμής. Τα σύνολα εκπαίδευσης χρησιμοποιούνται για την εκπαίδευση του μοντέλου, ενώ για την παρακολούθηση της απόδοσης και την εξαγωγή συμπερασμάτων χρησιμοποιείται το σύνολο επικύρωσης που ρυθμίζει τις υπερπαραμέτρους.

- Για τη βελτιστοποίηση της απόδοσης του μοντέλου υπάρχει το προαιρετικό βήμα που θα μπορούσαμε να δοκιμάσουμε την αλλαγή διάφορων υπερπαραμέτρων όπως είναι ο ρυθμός μάθησης, το μέγεθος της παρτίδας και την αλλαγή στην αρχιτεκτονική του μοντέλου.

- Αξιολόγηση του μοντέλου που γίνεται χρησιμοποιώντας χαρακτηριστικά για την απόδοση και την πρόοδο του μοντέλου όπως είναι η ακρίβεια, η ανάκλαση, το F1-score και η μέση ακρίβεια (mAP)

- Δοκιμές του μοντέλου αφού έχει ολοκληρωθεί η εκπαίδευσή του και έχει επικυρωθεί, με αποτέλεσμα να μπορεί να αξιολογηθεί η απόδοσή του στο σύνολο των δεδομένων δοκιμής, στη συγκεκριμένη περίπτωση να μπορεί να ανιχνεύει το αντικείμενο το οποίο έχουμε αποφασίσει να εξετάσουμε.

- Τέλος υπάρχει η μετα-επεξεργασία των δεδομένων με σκοπό να βελτιωθούν οι ανιχνευόμενες περιοχές να αφαιρεθούν τα αποτελέσματα που είναι ψευδώς θετικά με σκοπό τη βελτίωση της ανίχνευσης.

4.1.3 Ψευδοκώδικας αλγορίθμου Faster R-CNN

Απεικόνιση 4.1: Ψευδοκώδικας για τον αλγόριθμο Faster R-CNN.

```
# Import necessary libraries/modules
import os
import shutil
import numpy as np
from tensorflow.keras.applications.resnet50 import ResNet50
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.resnet50 import
   preprocess_input, decode_predictions
import time

# Start a timer
start_time = time.time()

# Define source and destination folders
source_folder = '/path/to/source/folder'
destination_folder = '/path/to/destination/folder'

# Load a pre-trained ResNet50 model
model = ResNet50(weights='imagenet')

# Loop through images in the source folder
for filename in os.listdir(source_folder):
    # Load the image
    img_path = os.path.join(source_folder, filename)
    img = image.load_img(img_path, target_size=(224, 224))
```

```
    # Preprocess the image
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0)
    x = preprocess_input(x)

    # Predict the class of the image
    preds = model.predict(x)
    top_predictions = decode_predictions(preds, top=5)[0]  #
       Get the top 5 predicted classes

    # Initialize variables to store predicted class
    predicted_class = None

    # Check if "cat" or "dog" is in the top predicted classes
    for prediction in top_predictions:
        class_name = prediction[1].lower()
        if 'cat' in class_name or 'dog' in class_name or '
           elephant' in class_name:
            predicted_class = class_name
            break

    # If "cat" or "dog" is found in the recognized class, move
       the image to the appropriate folder
    if predicted_class:
        class_folder = os.path.join(destination_folder,
           predicted_class)
        if not os.path.exists(class_folder):
            os.makedirs(class_folder)
        shutil.move(img_path, os.path.join(class_folder,
           filename))

# Calculate the elapsed time
```

```
end_time = time.time()
elapsed_time = end_time - start_time
elapsed_time = elapsed_time / 60

# Print the execution time
print(f"Execution time: {elapsed_time} minutes")
```

4.2 Ο αλγόριθμος YOLO

4.2.1 Γενικά

Ο αλγόριθμος YOLO είναι ευρεία χρησιμοποιούμενος για την ανίχνευση αντικειμένων που ανήκουν σε διαφορετικές κατηγορίες και αποτελεί μια μέθοδο βαθιάς μάθησης που παρέχει ανίχνευση δεδομένων σε πραγματικό χρόνο κάνοντας χρήση νευρωνικά δίκτυα. Ο YOLO έχει τα εξής χαρακτηριστικά: τα αποτελέσματα που παρέχει έχουν μεγάλη ακρίβεια, εξάγονται σε γρήγορο χρονικό διάστημα και έχει τη δυνατότητα εκμάθησης που του επιτρέπει να μαθαίνει τις αναπαραστάσεις των αντικειμένων για μελλοντικές ανιχνεύσεις. Ο αλγόριθμος δέχεται μια εικόνα με σκοπό να αναγνωρίσει αντικείμενα σε αυτή και στη συνέχεια την διαιρεί σε πλέγμα κελιών. Κάθε κελί από αυτά που δημιουργούνται περιέχει ένα σταθερό αριθμό οροθετημένων πλαισίων με τη μορφή πρόβλεψης μαζί με την πιθανότητα να εμφανιστεί το συγκεκριμένο αντικείμενο για το οποίο ερευνούμε στο οριοθετημένο πλαίσιο που έχει δημιουργηθεί και την κατηγορία του αντικειμένου. Τέλος, ο αλγόριθμος YOLO εφαρμόζει μια μη μέγιστη καταστολή, με σκοπό να αφαιρεθούν οι διπλές ανιχνεύσεις και εξάγει το τελικό σύνολο των οροθετημένων κουτιών. Το σύνολο των δεδομένων που θα χρησιμοποιηθούν με σκοπό την εκπαίδευση και τον προσδιορισμό της ύπαρξης ενός αντικειμένου ή όχι από τον αλγόριθμο YOLO χρειάζεται να έχει τη μορφή εικόνας και κάθε εικόνα να περιέχει το αντίστοιχο αρχείο κειμένου που περιέχει πληροφορίες για την έκταση και τη θέση του αντικειμένου στην κάθε απεικόνιση. Επιπροσθέτως το σύνολο δεδομένων χρειάζεται να είναι μεγάλο έτσι ώστε η εκπαίδευση του αλγορίθμου να πραγματοποιηθεί αποτελεσματικά.

4.2.2 Τρόπος Υλοποίησης αλγορίθμου YOLO

Ο αλγόριθμος YOLO για την αναγνώριση αντικειμένων σε εικόνες ακολουθεί την εξής διαδικασία:

- Συλλογή δεδομένων από ένα σύνολο δεδομένων που περιέχει διαφορετικά είδη του ίδιου αντικειμένου που τίθεται υπό αναγνώριση.
- Σχολιασμός δεδομένων έτσι ώστε στις περιοχές στις οποίες συναντάτε η εμφάνιση του αντικειμένου να δημιουργηθούν οι περιοχές ενδιαφέροντος ROI και να παραχθούν οριοθετημένα πλαίσια γύρω από τους όγκους .
- Προεπεξεργασία δεδομένων δηλαδή το σύνολο το δεδομένων που διαθέτουμε για εκπαίδευση προετοιμάζεται κατάλληλα αλλάζοντας το μέγεθος των φωτογραφιών σε ένα σταθερό κοινό μέγεθος, κανονικοποίηση των τιμών, των εικονοστοιχείων και τέλος σε αυτό το βήμα γίνεται διαχωρισμός του συνόλου δεδομένων σε τρία νέα σύνολα της εκπαίδευσης, της επικύρωσης και της δοκιμής.
- Επιλογή μοντέλου με κατάλληλη αρχιτεκτονική YOLO το οποίο εξαρτάται από τους υπολογιστικούς πόρους που διαθέτουμε.
- Εκπαίδευση του μοντέλου χρησιμοποιώντας το σχολιασμένο σύνολο δεδομένων.
- Ρύθμιση του ρυθμού εκμάθησης, του μεγέθους της δέσμης και του κουτιού αγκύρωσης έτσι ώστε να βελτιστοποιηθεί η απόδοση του αλγορίθμου.
- Αξιολόγηση της απόδοσης του μοντέλου που εκπαιδεύσαμε χρησιμοποιώντας ένα διαφορετικό σύνολο δεδομένων από αυτό που χρησιμοποιήσαμε για την εκπαίδευση, σύνολο επικύρωσης, λαμβάνοντας μετρήσεις όπως είναι η ακρίβεια, η ανάκλαση, το F1-score και η μέση ακρίβεια (mAP) του YOLO μοντέλου που χρησιμοποιήσαμε.
- Τέλος γίνεται περαιτέρω δοκιμή του μοντέλου σε ένα δοκιμαστικό σύνολο δεδομένων με σκοπό την αξιολόγησή του και τη λήψη μετρήσεων σε νέα δεδομένα που δεν περιέχονταν στο αρχικό σύνολο δεδομένων που χρησιμοποιήσαμε για το YOLO μοντέλο.

4.2.3 Ψευδοκώδικας αλγορίθμου YOLO

Απεικόνιση 4.2: Ψευδοκώδικας για τον αλγόριθμο YOLO.

```
# Import necessary libraries/modules
import os
import shutil
from PIL import Image
import torch
import time

# Initialize lists to store true and predicted labels
true_labels = []
predicted_labels = []

# Start a timer
start_time = time.time()

# Load the YOLOv5 model
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

# Set the device to use (CPU or GPU)
device = torch.device('cuda' if torch.cuda.is_available() else
    'cpu')

# Define the source and destination folders
source_folder = '/path/to/source/folder'
destination_folder = '/path/to/destination/folder'

# Define a set of animal labels you want to detect
animal_labels = {15, 16, 20}

# Loop through all the images in the source folder
```

```
for filename in os.listdir(source_folder):
    # Load the image
    image_path = os.path.join(source_folder, filename)
    image = Image.open(image_path)

    # Perform object detection on the image
    results = model(image)

    # Get the predicted labels and bounding boxes
    labels = results.xyxy[0][:, -1].cpu().numpy()
    boxes = results.xyxy[0][:, :-1].cpu().numpy()

    # Iterate over the predicted objects
    for label, box in zip(labels, boxes):
        # Check if the object is an animal with a label in the
           set
        if label in animal_labels:
            # Get the animal name from the label
            animal_name = results.names[int(label)]

            # Create the destination folder if it doesn't exist
            animal_folder = os.path.join(destination_folder,
               animal_name)
            if not os.path.exists(animal_folder):
                os.makedirs(animal_folder)

            # Move the image to the appropriate folder
            shutil.copy(image_path, os.path.join(animal_folder,
                filename))

        # Store true and predicted labels for further
           processing/metrics
```

```
        true_labels.append(true_label)
        predicted_labels.append(predicted_label)

# Calculate metrics for class 15, class 16, and class 20
# Calculate other desired metrics if needed

# Calculate execution time
end_time = time.time()
elapsed_time = end_time - start_time
elapsed_time = elapsed_time / 60

# Print the execution time
print(f"Execution time: {elapsed_time} minutes")
```

4.3 Ο αλγόριθμος DES

4.3.1 Γενικά

Ο αλγόριθμος κρυπτογράφησης DES ανήκει στη κατηγορία αλγορίθμων συμμετρικής κρυπτογράφησης μπλοκ. Λειτουργεί με τον εξής τρόπο: λαμβάνει ένα σύνολο δεδομένων σε μορφή μπλοκ απλού κειμένου σε συνδυασμό με ένα μυστικό κλειδί κρυπτογράφησης, έτσι ώστε να επιτευχθεί η μετατροπή του απλού αυτού κειμένου σε κρυπτογράφημα. Αν και ο αρχικός σχεδιασμός του αλγορίθμου DES ήταν για κρυπτογράφηση κειμένου μπορεί να χρησιμοποιηθεί και για κρυπτογράφηση εικόνων μετατρέποντάς τες πρώτα σε δυαδικές ακολουθίες. Η χρησιμότητά του όπως και κάθε αλγορίθμου κρυπτογράφησης είναι η διαφύλαξη των δεδομένων αποτρέποντας τρίτους να έχουν πρόσβαση σε αυτά ακόμη και αν αυτή η πρόσβαση είναι η απλή ανάγνωσή τους. Για να επιτευχθεί η κρυπτογράφηση στο σύνολο των εικόνων που διαθέτουμε δεν χρειάζεται να είναι κάποιου συγκεκριμένου είδους παρά μόνο να έχει γίνει η μετατροπή τους σε δυαδική μορφή. Οπότε είναι δυνατή και η κρυπτογράφηση οποιασδήποτε εικόνας.

4.3.2 Τρόπος Υλοποίησης αλγορίθμου DES

Ο αλγόριθμος DES για την κρυπτογράφηση εικόνων ακολουθεί την εξής διαδικασία:

- Εύρεση συνόλου δεδομένων με φωτογραφίες
- Μετατροπή της εικόνας σε δυαδική μορφή το οποίο πραγματοποιείται με τη μετατροπή της τιμής κάθε εικονοστοιχείου σε δυαδική μορφή.
- Χωρισμός της δυαδικής εικόνας σε μπλοκ δεδομένων με σταθερό μέγεθος το οποίο χρειάζεται να είναι πολλαπλάσιο των 64bit.
- Δημιουργία κλειδιού κρυπτογράφησης DES το οποίο έχει μέγεθος 56-bit και χρησιμοποιείται για να παραχθούν 16 υποκλειδιά ένα για κάθε γύρο κρυπτογράφησης που προκύπτουν μέσα από μια διαδικασία προγραμματισμού κλειδιών.
- Το μπλοκ του δυαδικού αρχείου που παράχθηκε έχει μήκος 64-bit οπότε υποβάλλεται σε μια αρχική μεταβολή αναδιατάσσοντας αυτά τα bit σύμφωνα με έναν σταθερό πίνακα του αλγορίθμου DES.
- Χρησιμοποίηση ενός Feistel δικτύου που αποτελείται από 16 γύρους και σε κάθε γύρο ακολουθούνται τα εξής πέντε βήματα στο ήμισυ των δεδομένων του κάθε μπλοκ. Πρώτον γίνεται επέκταση των 32-bit του κάθε μπλοκ σε 48-bit με την αναπαραγωγή ορισμένων bit. Δεύτερον γίνεται μίξη κλειδιών δηλαδή τα δεδομένα του μπλοκ που έχουν υποστεί επέκταση συνδυάζονται με ένα υποκλειδί συγκεκριμένου γύρου κρυπτογράφησης του αρχικού κλειδιού χρησιμοποιώντας μια XOR. Τρίτον το νέο αποτέλεσμα των 48-bit που προκύπτει διαιρείται σε 8 ομάδες που η κάθε μια αποτελείται από 6-bit και στη συνέχεια χρησιμοποιώντας έναν προκαθορισμένο πίνακα που ονομάζεται κουτί υποκατάστασης γίνεται αντικατάσταση των έξι αυτών bit με τέσσερα που έχει ως αποτέλεσμα να προστίθεται ένα μη γραμμικό στοιχείο στην κρυπτογράφηση.Τέταρτον τα 32-bit που δημιουργήθηκαν από το προηγούμενο κουτί υποκατάστασης χρησιμοποιούνται σε συνδυασμό με έναν άλλον προκαθορισμένο πίνακα έτσι ώστε να γίνει αντιμετάθεση σε αυτά. Πέμπτον και

τελευταίο το αποτέλεσμα της προηγούμενης αντιμετάθεσης γίνεται XOR με το υπόλοιπο μισό μπλοκ του προηγούμενου γύρου κρυπτογράφησης.

- Αφού ολοκληρωθούν όλοι οι γύροι του Feistel δικτύου τα δύο μισά μπλοκ ανταλλάσσονται.
- Εφαρμόζεται αντιστροφή στην αρχική αντιμετάθεση των μπλοκ
- Τέλος Παράγεται το κρυπτογράφημα και η ίδια διαδικασία επαναλαμβάνεται για όλα τα μπλοκ της εικόνας που έχουν παραχθεί.

4.3.3 Ψευδοκώδικας

Απεικόνιση 4.3: Ψευδοκώδικας για τον αλγόριθμο DES.

```
# Import necessary libraries/modules
from Crypto.Cipher import DES
from Crypto.Random import get_random_bytes
import os
import time

# Start a timer
start_time = time.time()

# Encryption function for a dataset of images using DES
def encrypt_images(dataset_folder, output_folder, key):
    # Create a DES cipher object with the provided key
    cipher = DES.new(key, DES.MODE_ECB)

    # Iterate through the images in the dataset folder
    for filename in os.listdir(dataset_folder):
        if filename.endswith(".jpg"):  # Assuming your dataset
           contains JPEG images
            input_image_path = os.path.join(dataset_folder,
               filename)
```

```
            output_image_path = os.path.join(output_folder,
               filename.replace('.jpg', '.enc'))

            # Read the plaintext image
            with open(input_image_path, 'rb') as f:
                plaintext = f.read()

            # Pad the plaintext to match the block size of DES
            plaintext += b'\x00' * (8 - (len(plaintext) % 8))

            # Encrypt the plaintext
            ciphertext = cipher.encrypt(plaintext)

            # Write the encrypted image to the output folder
            with open(output_image_path, 'wb') as f:
                f.write(ciphertext)

# Decryption function for a dataset of images using DES
def decrypt_images(encrypted_folder, output_folder, key):
    # Create a DES cipher object with the provided key
    cipher = DES.new(key, DES.MODE_ECB)

    # Iterate through the encrypted images in the folder
    for filename in os.listdir(encrypted_folder):
```

4.4 Ο αλγόριθμος AES

4.4.1 Γενικά

Ο αλγόριθμος AES ανήκει στην κατηγορία των αλγορίθμων συμμετρικής κρυπτογράφησης που χρησιμοποιούν μπλοκ δεδομένων και υποστηρίζει μήκη κλειδιών 128-bit, 192-bit και 256-bit. Επειδή ανήκει στην κατηγορία αλγορίθμων που χρησιμοποιούν μπλοκ δεδομένων ο AES λαμβάνει ένα σύνολο δεδομένων σε μορφή μπλοκ

απλού κειμένου σε συνδυασμό με ένα μυστικό κλειδί κρυπτογράφησης έτσι ώστε να επιτευχθεί η μετατροπή του απλού αυτού κειμένου σε κρυπτογράφημα. Ο αλγόριθμος αυτός μπορεί να χρησιμοποιηθεί τόσο για την κρυπτογράφηση κειμένου όσο και στην κρυπτογράφηση εικόνων με την προϋπόθεση να έχουν την κατάλληλη μορφή, δηλαδή να αναπαριστώνται οι εικόνες ως ακολουθία από bytes. Οπότε μπορεί να χρησιμοποιηθεί ο συγκεκριμένος αλγόριθμος και για την κρυπτογράφηση εικόνων οποιουδήποτε περιεχομένου.

4.4.2 Τρόπος Υλοποίησης αλγορίθμου AEs

Ο αλγόριθμος AES για την κρυπτογράφηση εικόνων ακολουθεί την εξής διαδικασία:

- Εύρεση συνόλου δεδομένων με επιθυμητές φωτογραφίες
- Αναπαράσταση κάθε εικόνας σε μορφή ακολουθίας bytes χρησιμοποιώντας μια κατάλληλη βιβλιοθήκη επεξεργασίας εικόνας όπως είναι η Pillow ή η OpenCV οι οποίες διαβάζουν κάθε εικόνα και τη μετατρέπουν σε ένα byte πίνακα.
- Ο αλγόριθμος AES λειτουργεί με μπλοκ σταθερού μεγέθους το οποίο είναι είτε 128-bit είτε 16-byte. Οπότε στην περίπτωση που το μέγεθος της εικόνας δεν είναι πολλαπλάσιο του μεγέθους του μπλοκ, προστίθεται στο τελευταίο μπλοκ της εικόνας ο απαιτούμενος αριθμός bit έτσι ώστε να υπάρχει αυτή η συμβατότητα μεγέθους.
- Ο αλγόριθμος AES εφαρμόζεται σε κάθε μπλοκ της εικόνας ξεχωριστά και χρησιμοποιεί ένα μυστικό κλειδί που το μέγεθός του μπορεί να είναι 128-bit, 192-bit ή 256-bit ανάλογα με το επίπεδο ασφάλειας που θέλουμε να επιτύχουμε. Το κλειδί αυτό χρησιμοποιείται για να παραχθεί ένα σύνολο από κλειδιά που θα χρησιμοποιηθούν σε κάθε γύρο κρυπτογράφησης. Το πλήθος αυτών των γύρων κρυπτογράφησης έχει διαφορετική τιμή και εξαρτάται από το μήκος του κλειδιού κρυπτογράφησης που χρησιμοποιεί ο αλγόριθμος AES. Δηλαδή για μήκος κλειδιού ίσο με 128-bit ο αριθμός των γύρων κρυπτογράφησης είναι δέκα, για μήκος κλειδιού 192-bit και 256-bit το πλήθος των γύρων κρυπτογράφησης είναι δώδεκα και δεκατέσσερα αντίστοιχα.

- Στον πρώτο γύρο κρυπτογράφησης (γύρος 0), ο αλγόριθμος AES λειτουργεί σε μπλοκ δεδομένων με μέγεθος συνήθως 128-bit και πραγματοποιεί την πράξη XOR ανάμεσα σε ένα μπλοκ της εικόνας που έχουμε επιλέξει να κρυπτογραφήσουμε τη δεδομένη χρονική στιγμή και του κλειδιού του πρώτου γύρου κρυπτογράφησης.

- Στους υπόλοιπους γύρους κρυπτογράφησης που ακολουθούν μέχρι και τον N-1 πραγματοποιούνται τέσσερις βασικοί μετασχηματισμοί οι οποίοι είναι: ο SybBytes που αντικαθιστά κάθε byte του μπλοκ με ένα άλλο που προέρχεται από έναν σταθερό πίνακα αντικατάστασης. Ο ShiftRows που μετατοπίζει τις γραμμές του μπλοκ. Ο MixColums που αλλάζει τη θέση των στηλών του μπλοκ με σκοπό την περαιτέρω διασπορά δεδομένων και ο AddRoundKey που πραγματοποιεί τη λογική πράξη XOR ανάμεσα στο μπλοκ της εικόνας και στο κλειδί του τρέχοντος γύρου κρυπτογράφησης.

- Στον τελικό γύρο κρυπτογράφησης εφαρμόζονται οι μετασχηματισμοί που εφαρμόζονται και στους υπόλοιπους γύρους κρυπτογράφησης με τη διαφορά ότι ο μετασχηματισμός MixColumns παραλείπεται.

- Τέλος παράγεται ως έξοδος το κρυπτογράφημα και η ίδια διαδικασία επαναλαμβάνεται για όλα τα μπλοκ της εικόνας που έχουν παραχθεί.

4.4.3 Ψευδοκώδικας αλγορίθμου AES

Απεικόνιση 4.4: Ψευδοκώδικας για τον αλγόριθμο AES.

```
# Import necessary libraries/modules
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import os
import time

# Start a timer
start_time = time.time()

# Encryption function for a dataset of images using AES
```

```
def encrypt_images(dataset_folder, output_folder, key):
    # Create an AES cipher object with the provided key
    cipher = AES.new(key, AES.MODE_ECB)

    # Iterate through the images in the dataset folder
    for filename in os.listdir(dataset_folder):
        if filename.endswith(".jpg"):  # Assuming your dataset
           contains JPEG images
            input_image_path = os.path.join(dataset_folder,
               filename)
            output_image_path = os.path.join(output_folder,
               filename.replace('.jpg', '.enc'))

            # Read the plaintext image
            with open(input_image_path, 'rb') as f:
                plaintext = f.read()

            # Pad the plaintext to match the block size of AES
               (16 bytes)
            plaintext += b'\x00' * (16 - (len(plaintext) % 16))

            # Encrypt the plaintext
            ciphertext = cipher.encrypt(plaintext)

            # Write the encrypted image to the output folder
            with open(output_image_path, 'wb') as f:
                f.write(ciphertext)

# Decryption function for a dataset of images using AES
def decrypt_images(encrypted_folder, output_folder, key):
    # Create an AES cipher object with the provided key
    cipher = AES.new(key, AES.MODE_ECB)
```

```
    # Iterate through the encrypted images in the folder
    for filename in os.listdir(encrypted_folder):
        if filename.endswith(".enc"):
            input_image_path = os.path.join(encrypted_folder,
               filename)
            output_image_path = os.path.join(output_folder,
               filename.replace('.enc', '.jpg'))

            # Read the encrypted image
            with open(input_image_path, 'rb') as f:
                ciphertext = f.read()

            # Decrypt the ciphertext
            plaintext = cipher.decrypt(ciphertext)

            # Remove the padding from the plaintext
            plaintext = plaintext.rstrip(b'\x00')

            # Write the decrypted image to the output folder
            with open(output_image_path, 'wb') as f:
                f.write(plaintext)

# Example usage
key = get_random_bytes(16)  # Generate a 128-bit key for AES
dataset_folder = '/path/to/original/dataset'
encrypted_folder = '/path/to/encrypted/images'
decrypted_folder = '/path/to/decrypted/images'

os.makedirs(encrypted_folder, exist_ok=True)
os.makedirs(decrypted_folder, exist_ok=True)
```

```
# Encrypt the images
encrypt_images(dataset_folder, encrypted_folder, key)

# Decrypt the encrypted images
decrypt_images(encrypted_folder, decrypted_folder, key)

# Calculate execution time
end_time = time.time()
elapsed_time = end_time - start_time
elapsed_time = elapsed_time / 60

# Print the execution time
print(f"Execution time: {elapsed_time} minutes")
```

4.5 Ο αλγόριθμος BlowFish

4.5.1 Γενικά

Ο αλγόριθμος Blowfish ανήκει στην κατηγορία αλγορίθμων μπλοκ συμμετρικού κλειδιού και κρυπτογραφεί τα δεδομένα που επεξεργάζεται, κάνοντας χρήση ενός κλειδιού μεταβλητού μήκους, που το μέγεθός του κυμαίνεται από 32-bit μέχρι 448-bit. Το κλειδί αυτό χρησιμοποιείται για την αρχικοποίηση της κατάστασης του αλγορίθμου Blowfish και χωρίζεται σε μπλοκ που έχουν μέγεθος 8-bit το καθένα. Η αρχική κτάσταση του αλγορίθμου Blowfish αποτελείται από μια συστοιχία P (P-array) που είναι ένα σύνολο υποκλειδιών και έχει μέγεθος 64-bit και από τέσσερα πλαίσια S (S-boxes) που είναι πλαίσια αντικατάστασης και έχουν μέγεθος 32-bit. Ο αλγόριθμος Blowfish χρησιμοποιείται τόσο για την κρυπτογράφηση κειμένου όσο και για την κρυπτογράφηση εικόνων και βίντεο οποιαδήποτε κατηγορίας. Η μόνη προϋπόθεση που χρειάζεται να πληρείτε για την κρυπτογράφηση των εικόνων είναι τα δεδομένα να έχουν μετατραπεί σε δυαδική μορφή.

4.5.2 Τρόπος Υλοποίησης αλγορίθμου Blowfish

Ο αλγόριθμος Blowfish για την κρυπτογράφηση εικόνων ακολουθεί την εξής διαδικασία:

- Εύρεση συνόλου δεδομένων με φωτογραφίες
- Μετατροπή της κάθε εικόνας στην κατάλληλη μορφή για να είναι έτοιμη για κρυπτογράφηση που είναι η μεταποίησή της σε δυαδική αναπαράσταση χρησιμοποιώντας μια βιβλιοθήκη επεξεργασίας εικόνας που μπορεί να πραγματοποιήσει αυτή τη μετατροπή.
- Δημιουργία ενός μυστικού κλειδιού κρυπτογράφησης που θα χρησιμοποιήσει ο αλγόριθμος BlowFish με μέγεθος από 32-bits έως 448-bits.
- Ορισμός των κατάλληλων πινάκων P-array και S-boxes που χρησιμοποιούνται για την κρυπτογράφηση των δεδομένων από τον αλγόριθμο Blowfish.
- Σε περίπτωση που το μέγεθος της εικόνας που χρησιμοποιούμε δεν είναι πολλαπλάσιο του μεγέθους μπλοκ που χρησιμοποιεί ο αλγόριθμος BlowFish δηλαδή 64-bit χρειάζεται να συμπληρώσουμε τα δεδομένα με τον απαιτούμενο αριθμό bits κάνοντας χρήση ενός σχήματος συμπλήρωσης όπως είναι το PKCS7.
- Διαχωρισμός της κάθε εικόνας σε επιμέρους μπλοκ δεδομένων με μέγεθος 64-bit.
- Κρυπτογράφηση του κάθε μπλοκ της εικόνας με τον αλγόριθμο BlowFish που περιλαμβάνει συνήθως δεκαέξι γύρους με ανάμιξη και αντικατάσταση δεδομένων κάνοντας χρήση των P-array και των S-boxes, που έχουν οριστεί κατάλληλα σε προηγούμενο βήμα της διαδικασίας, με την οποία γίνεται η κρυπτογράφηση των εικόνων κάνοντας χρήση του αλγόριθμου BlowFish.
- Η διαδικασία αυτή επαναλαμβάνεται για όλα τα μπλοκ κάθε εικόνας μέχρι να κρυπτογραφηθούν όλα και κατ' επέκταση για κάθε φωτογραφία του συνόλου δεδομένων.
- Τέλος στην περίπτωση κρυπτογράφησης πολλαπλών μπλοκ ή εικόνων με το ίδιο κλειδί δημιουργείται ένα διάνυσμα αρχικοποίησης με τυχαία τιμή για

την εξασφάλιση της ασφάλειας των δεδομένων μας. Έτσι ο αλγόριθμος έχει ως έξοδο ένα κρυπτογραφημένο κείμενο που είναι ο συνδυασμός των αποθηκευμένων κρυπτογραφημένων μπλοκ των δεδομένων της εικόνας με διάνυσμα αρχικοποίησης.

4.5.3 Ψευδοκώδικας

Απεικόνιση 4.5: Ψευδοκώδικας για τον αλγόριθμο Blowfish.

```
# Import necessary libraries/modules
from Crypto.Cipher import Blowfish
from Crypto.Random import get_random_bytes
import os
import time

# Start a timer
start_time = time.time()

# Encryption function for a dataset of images using Blowfish
def encrypt_images(dataset_folder, output_folder, key):
    # Create a Blowfish cipher object with the provided key
    cipher = Blowfish.new(key, Blowfish.MODE_ECB)

    # Iterate through the images in the dataset folder
    for filename in os.listdir(dataset_folder):
        if filename.endswith(".jpg"):  # Assuming your dataset
           contains JPEG images
            input_image_path = os.path.join(dataset_folder,
               filename)
            output_image_path = os.path.join(output_folder,
               filename.replace('.jpg', '.enc'))

            # Read the plaintext image
            with open(input_image_path, 'rb') as f:
```

```
                plaintext = f.read()

            # Pad the plaintext to match the block size of
               Blowfish (8 bytes)
            plaintext += b'\x00' * (8 - (len(plaintext) % 8))

            # Encrypt the plaintext
            ciphertext = cipher.encrypt(plaintext)

            # Write the encrypted image to the output folder
            with open(output_image_path, 'wb') as f:
                f.write(ciphertext)

# Decryption function for a dataset of images using Blowfish
def decrypt_images(encrypted_folder, output_folder, key):
    # Create a Blowfish cipher object with the provided key
    cipher = Blowfish.new(key, Blowfish.MODE_ECB)

    # Iterate through the encrypted images in the folder
    for filename in os.listdir(encrypted_folder):
        if filename.endswith(".enc"):
            input_image_path = os.path.join(encrypted_folder,
               filename)
            output_image_path = os.path.join(output_folder,
               filename.replace('.enc', '.jpg'))

            # Read the encrypted image
            with open(input_image_path, 'rb') as f:
                ciphertext = f.read()

            # Decrypt the ciphertext
            plaintext = cipher.decrypt(ciphertext)
```

```
            # Remove the padding from the plaintext
            plaintext = plaintext.rstrip(b'\x00')

            # Write the decrypted image to the output folder
            with open(output_image_path, 'wb') as f:
                f.write(plaintext)

# Example usage
key = get_random_bytes(16)  # Generate a 128-bit key for
   Blowfish
dataset_folder = '/path/to/original/dataset'
encrypted_folder = '/path/to/encrypted/images'
decrypted_folder = '/path/to/decrypted/images'

os.makedirs(encrypted_folder, exist_ok=True)
os.makedirs(decrypted_folder, exist_ok=True)

# Encrypt the images
encrypt_images(dataset_folder, encrypted_folder, key)

# Decrypt the encrypted images
decrypt_images(encrypted_folder, decrypted_folder, key)

# Calculate execution time
end_time = time.time()
elapsed_time = end_time - start_time
elapsed_time = elapsed_time / 60

# Print the execution time
print(f"Execution time: {elapsed_time} minutes")
```

Κεφάλαιο 5

Υπολογιστική μελέτη

5.1 Αλγόριθμοι Αναγνώρισης Αντικειμένων

Για την υπολογιστική μελέτη επιλέχθηκαν να πραγματοποιηθούν μετρήσεις με τους αλγόριθμους Faster RCNN και YOLO.

- Τα δεδομένα των φωτογραφιών (dataset) όπου απεικονίζουν τα προς αναγνώριση αντικείμενα ήταν ελεύθερα για χρήση και έγινε λήψη τους από βιβλιοθήκες δεδομένων εικόνων όπως η kaggle. Τόσο τα παραπάνω δεδομένα της βιβλιοθήκης όσο και αυτά που αναζήτησα εγώ και πρόσθεσα, επεξεργάζοντας τα κατάλληλα, για τη βελτιστοποίηση της απόδοσης των αλγορίθμων προέρχονται από εικόνες google. Οι συλλογές των εικόνων οργανώθηκαν ανάλογα με το αντικείμενο που απεικονίζουν (σκύλος, γάτα, ελέφαντας, μήλο, πορτοκάλι, πινακίδα STOP κλπ) και τον αριθμό των εικόνων που περιέχουν. Έτσι έχουμε συλλογές δεδομένων 500, 1,500, 3,000 εικόνων. Τα δεδομένα για να αναγνωριστούν χαρακτηρίστηκαν με τις κατάλληλες ετικέτες.

- Κατά τη διάρκεια λήψης των μετρήσεων βελτιστοποιήθηκε η απόδοση των αλγορίθμων με τη χρήση των καταλληλότερων εκδόσεων βιβλιοθηκών της Python, keras, tensorflow και torch.

- Αν και ο χρόνος που απαιτείται για την ολοκλήρωση της αναγνώρισης αντικειμένων δεν αποτελεί μεταβλητή ποιοτικής απόδοσης του αλγορίθμου, θεωρήθηκε ότι πρέπει να περιλαμβάνεται στις μετρήσεις γιατί αποτελεί σημαντικό παράγοντα χρηστικότητας για την επιλογή του αλγορίθμου.

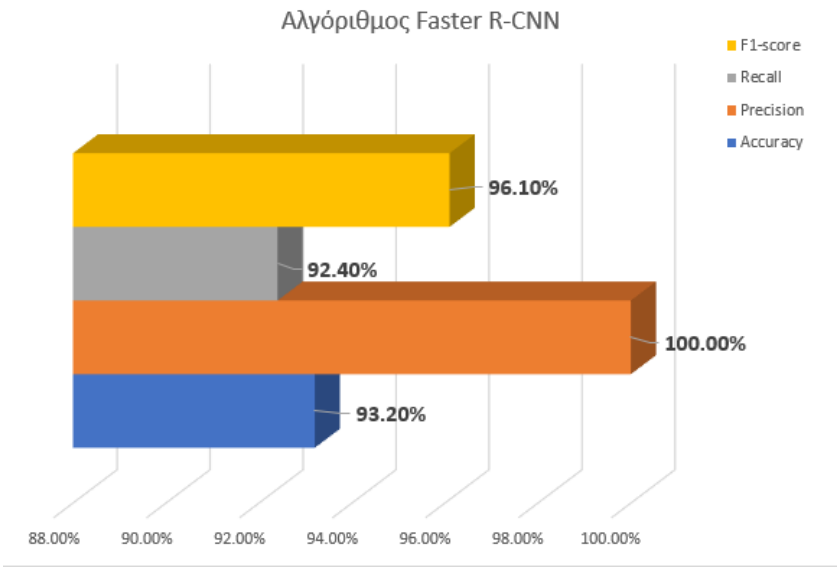
5.1.1 Ορισμοί μεταβλητών που χρησιμοποιούνται για τον υπολογισμό της αποδοτικότητας του μοντέλου

- Ακρίβεια(Accuracy): Η αναλογία του αριθμού των αληθινών θετικών προς τον συνολικό αριθμό των προβλέψεων $Accuracy = \frac{TN+TP}{TN+FP+TP+FN}$.
- Ανάκληση(Recall): Η αναλογία του αριθμού των αληθινών θετικών προς τον συνολικό αριθμό των θετικών προβλέψεων $Recall = \frac{TP}{TP+FN}$.
- Ακρίβεια (Precision): Ο λόγος του αριθμού των αληθινών θετικών προς τον συνολικό αριθμό των θετικών προβλέψεων $Precision = \frac{TP}{TP+FP}$.
- F1Score: Ο σταθμισμένος μέσος όρος της ακρίβειας και της ανάκλησης $F1 - score = 2 * \frac{Precision*Recall}{Precision+Recall}$. Οι τιμές κυμαίνονται από 0 έως 1 όπου το 1 σημαίνει μεγαλύτερη ακρίβεια.
- True-Positive (TP): Ο αριθμός των αληθινών θετικών που δημιουργούνται από το μοντέλο (αναγνωρίστηκαν σωστά).
- True-Negative (TN): Ο αριθμός των αληθινών αρνητικών που δημιουργούνται από το μοντέλο (αυτά τα αντικείμενα δεν αναγνωρίστηκαν).
- False-Positive (FP): Ο αριθμός των ψευδώς θετικών που δημιουργούνται από το μοντέλο (αναγνωρίστηκαν σωστά και έπρεπε να αναγνωριστούν λάθος).
- False-Negative (FN): Ο αριθμός των ψευδώς αρνητικών που δημιουργούνται από το μοντέλο (αναγνωρίστηκαν λάθος και έπρεπε να αναγνωριστούν σωστά).

5.1.2 Μετρήσεις αλγορίθμου Faster RCNN

- Μοντέλο: ResNet50
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-1 (γάτα)

Στον Πίνακα 5.1 και στο Σχήμα 5.1 παρουσιάζονται τα αποτελέσματα των μετρήσεων για τον αλγόριθμο Faster R-CNN σε ένα σύνολο δεδομένων ομοειδών αντικειμένων 500 εικόνων, Αντικείμενο-1. Τα αποτελέσματα των παραπάνω μετρήσεων μπορούν να χρησιμοποιηθούν για να εξάγουμε το συμπέρασμα ότι η μέτρηση του



Σχήμα 5.1: Μέτρηση αλγορίθμου Faster RCNN

Accuracy είναι κατά 10 τις εκατό σχεδόν πιο υψηλή σε σχέση με την αντίστοιχη μέτρηση για το σύνολο δεδομένων των ομοειδών αντικειμένων του Αντικείμενου-2. Και οι υπόλοιπες μετρικές παρουσιάζουν καλύτερα ποσοστά στο σύνολο δεδομένων του Αντικειμένου-1. Το μόνο το οποίο είναι σταθερό και στα δύο είναι ο χρόνος που χρειάστηκε ο αλγόριθμος για να εκτελέσει την αναγνώριση στα δύο σύνολα δεδομένων, το οποίο είναι λογικό καθώς τα δύο σύνολα δεδομένων είχαν το ίδιο μέγεθος.

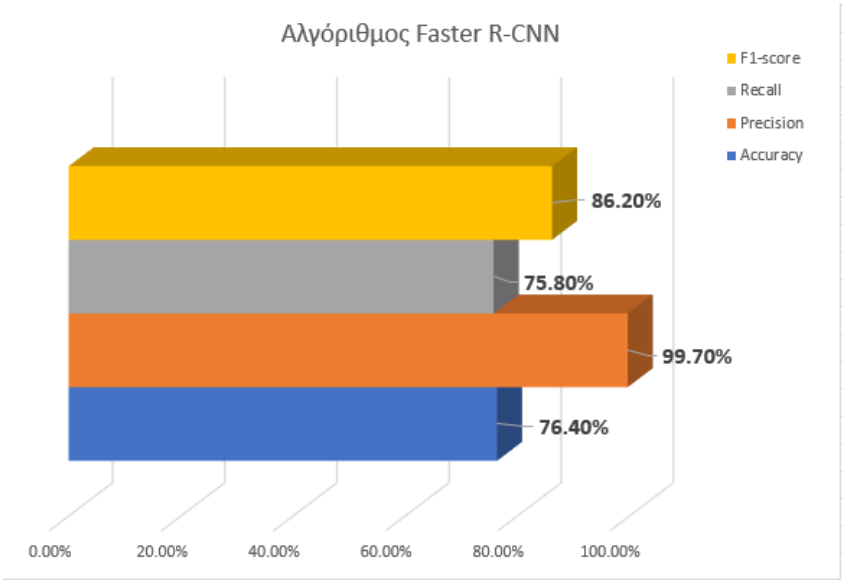
Πίνακας 5.1: Μετρικές αλγορίθμου Faster RCNN.

Μετρική	**Τιμή**
Accuracy	93.20
Precision	100.00
Recall	92.40
F1-score	96.10

Χρόνος εκτέλεσης: 1.33 λεπτά

- Μοντέλο: ResNet50
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-2 (σκύλος)

Για να πραγματοποιηθούν οι μετρήσεις που απεικονίζονται στον Πίνακα 5.2 και στο Σχήμα 5.2 του αλγορίθμου Faster R-CNN, μελετήθηκε ένα σύνολο δεδομένων 500 εικόνων ομοειδών αντικειμένων που περιείχε το Αντικείμενο-2. Κάνοντας μια



Σχήμα 5.2: Μέτρηση αλγορίθμου Faster RCNN

σύγκριση των αποτελεσμάτων, με αυτά που προκύπτουν από το σύνολο δεδομένων του Αντικειμένου-3, συμπεραίνουμε ότι η απόδοση του αλγορίθμου υστερεί στο σύνολο δεδομένων Αντικειμένου-2 για όλες τις μετρικές σε μεγάλο βαθμό εκτός βέβαια από αυτή του Precision που είναι σχεδόν η ίδια, καθώς επίσης και ο χρόνος εκτέλεσης του αλγορίθμου στις δυο περιπτώσεις.

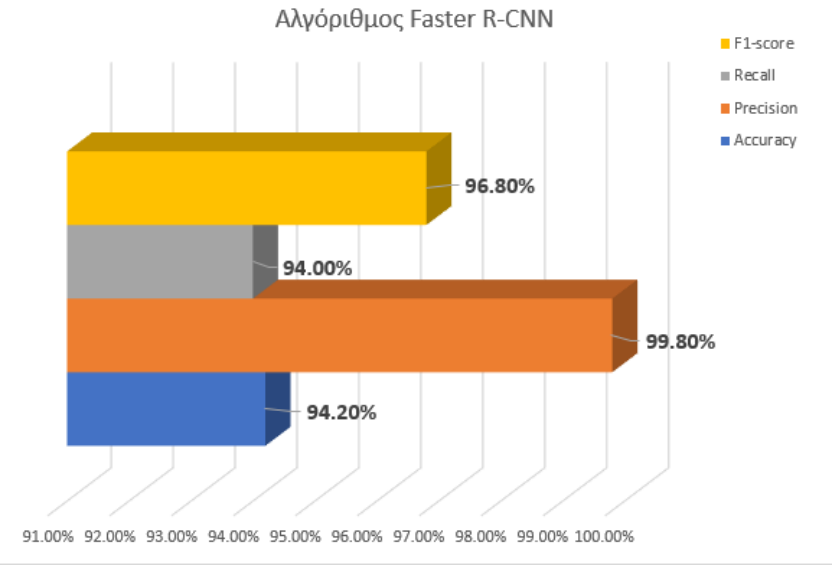
Πίνακας 5.2: Μετρικές αλγορίθμου Faster RCNN.

Μετρική	**Τιμή**
Accuracy	76.40
Precision	99.7
Recall	75.80
F1-score	86.20

Χρόνος εκτέλεσης : 1.32 λεπτά

- Μοντέλο: ResNet50
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-3 (ελέφαντας)

Ο Πίνακας 5.3 και το Σχήμα 5.3 περιέχουν τα αποτελέσματα που προκύπτουν από τη μέτρηση του Accuracy, του Recall, του Precision και του F1-score για τον αλγόριθμο Faster R-CNN, εκτελεσμένο σε ένα σύνολο δεδομένων που περιείχε μόνο το Αντικείμενο-3 και είχε μέγεθος 500 εικόνες. Παρατηρούμε ότι κάνοντας χρήση αυτού του συνόλου δεδομένων επιτυγχάνεται η καλύτερη απόδοση σε σχέση με αυτές



Σχήμα 5.3: Μέτρηση αλγορίθμου Faster RCNN

που προέκυψαν στην μελέτη των υπόλοιπον δύο συνόλων δεδομένων που το καθένα περιείχε μόνο το Αντικείμενο-1 ή το Αντικείμενο-2 αντίστοιχα. Οι μόνες μετρήσεις οι οποίες έχουν μικρή απόκλιση από σύνολο δεδομένων σε σύνολο δεδομένων είναι αυτή του Precision και του χρόνου.

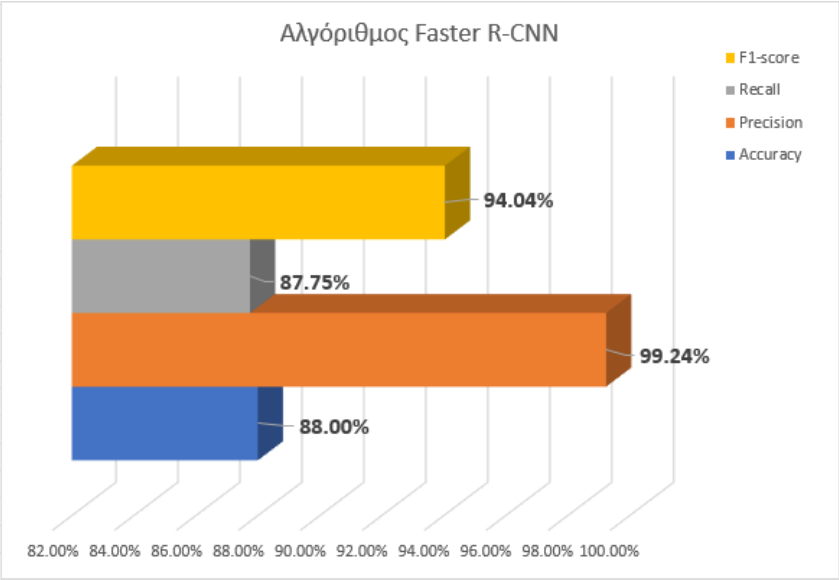
Πίνακας 5.3: Μετρικές αλγορίθμου Faster RCNN.

Μετρική	**Τιμή**
Accuracy	94.20
Precision	99.80
Recall	94.00
F1-score	96.80

Χρόνος εκτέλεσης: 1.34 λεπτά

- Μοντέλο: ResNet50
- Dataset: 1,500 εικόνες Αντικείμενο-1 500, Αντικείμενο-2 500, Αντικείμενο-3 500.

Στη συνέχεια χρησιμοποιώντας τα τρία σύνολα δεδομένων, Αντικείμενο-1, Αντκείμενο-2 και Αντικείμενο-3, δημιουργήσαμε ένα κοινό σύνολο δεδομένων με μέγεθος 1,500 εικόνων. Ύστερα από χρήση του συγκεκριμένου συνόλου δεδομένων για την εκτέλεση του αλγορίθμου Faster R-CNN προέκυψαν οι μετρήσεις που παρουσιάζονται στον Πίνακα 5.4, καθώς επίσης και στο Σχήμα 5.4. Κάνοντας μια σύγκριση τιμών



Σχήμα 5.4: Μέτρηση αλγορίθμου Faster RCNN

με τις τιμές του ίδιου αλγόριθμου αλλά σε σύνολο δεδομένων μικρότερου μεγέθους δηλαδή 500 εικόνων που περιέχει μόνο το Αντικείμενο-1, διαπιστώνουμε ότι το Accuracy μειώνεται κατά ένα ποσοστό 2 περίπου τις εκατό αλλά οι τιμές για το Precision και για το F1-score παρουσιάζουν ακόμη μεγαλύτερη μείωση της τάξης των 5 τις εκατό. Ο χρόνος εκτέλεσης του αλγορίθμου αυξάνεται, το οποίο είναι λογικό, διότι το μέγεθος του συνόλου δεδομένων γίνεται κατά τρεις φορές μεγαλύτερο.

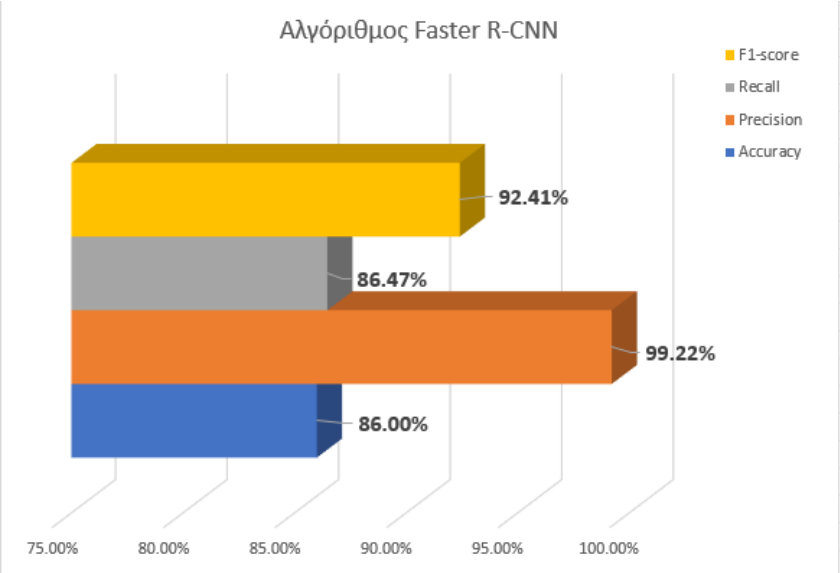
Πίνακας 5.4: Μετρικές αλγορίθμου Faster RCNN.

Μετρική	**Τιμή**
Accuracy	88.00
Precision	99.24
Recall	87.75
F1-score	94.04

Χρόνος εκτέλεσης : 3.78 λεπτά

- Μοντέλο: ResNet50
- Dataset: 3,000 εικόνες Αντικείμενο-1 500, Αντικείμενο-2 500, Αντικείμενο-3 500 και 1,500 εικόνες άλλων αντικειμένων (διαφορετικά ζώα).

Τέλος, έγινε η μέτρηση της αποδοτικότητας του αλγορίθμου Faster R-CNN σε ένα σύνολο δεδομένων με μέγεθος 3,000 εικόνων. Ήταν η σύνθεση των 1,500 εικόνων των συνόλων δεδομένων, Αντικείμενο-1, Αντικείμενο-2, Αντικείμενο-3, με ένα σύνολο



Σχήμα 5.5: Μέτρηση αλγορίθμου Faster RCNN

δεδομένων 1,500 εικόνων στο οποίο δεν συμπεριλαμβάνονται τα αντικείμενα αυτά, αλλά άλλα αντικείμενα. Τα αποτελέσματά του βρίσκονται στον Πίνακα 5.5 και στο Σχήμα 5.5. Ύστερα από σύγκριση προκύπτει ότι το σύνολο δεδομένων των 3,000 εικόνων δεν έχει τόσο καλά αποτελέσματα όσο το σύνολο δεδομένων των 1,500 εικόνων, οπότε η αύξηση του μεγέθους του συνόλου των φωτογραφιών με την προσθήκη άλλων εικόνων είχε αρνητική επίπτωση τις μετρήσεις μας.

Πίνακας 5.5: Μετρικές αλγορίθμου Faster RCNN.

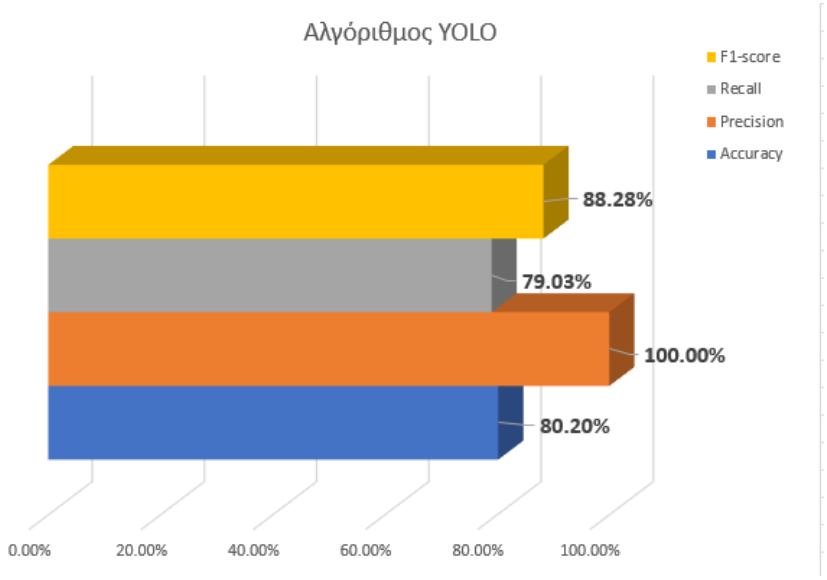
Μετρική	**Τιμή**
Accuracy	86.00
Precision	99.22
Recall	86.47
F1-score	92.41

Χρόνος εκτέλεσης : 7.6 λεπτά

5.1.3 Μετρήσεις αλγορίθμου YOLO

- Μοντέλο: YOLOv5
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-1 (γάτα)

Στον Πίνακα 5.6 και στο Σχήμα 5.6 παρουσιάζονται τα αποτελέσματα των μετρήσεων για τον αλγόριθμο YOLOV5 σε ένα σύνολο δεδομένων ομοειδών αντικειμένων 500 εικόνων, Αντικείμενο-1. Τα αποτελέσματα των παραπάνω μετρήσεων μπο-



Σχήμα 5.6: Μέτρηση αλγορίθμου YOLO

ρούν να χρησιμοποιηθούν για να εξάγουμε το συμπέρασμα ότι το σύνολο δεδομένων του Αντικειμένου-1 έχει χειρότερα αποτελέσματα σε σχέση με την εκτέλεση του ίδιου αλγορίθμου στο σύνολο δεδομένων του συνόλου δεδομένων του Αντικειμένου-2. Η μόνη μέτρηση που παραμένει σταθερή και στα δύο σύνολα δεδομένων, είναι ο χρόνος που χρειάστηκε ο αλγόριθμος για να εκτελέσει την αναγνώριση, το οποίο είναι λογικό καθώς τα δύο σύνολα δεδομένων είχαν το ίδιο μέγεθος.

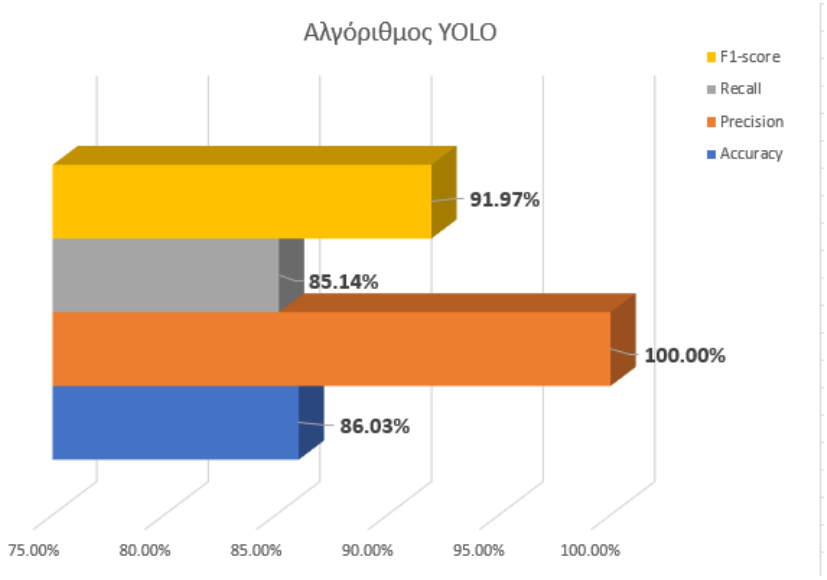
Πίνακας 5.6: Μετρικές αλγορίθμου YOLO.

Μετρική	**Τιμή**
Accuracy	80.2
Precision	100.00
Recall	79.03
F1-score	88.28

Χρόνος εκτέλεσης: 0.30 λεπτά

- Μοντέλο: YOLOv5
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-2 (σκύλος)

Για να πραγματοποιηθούν οι μετρήσεις που απεικονίζονται στον Πίνακα 5.7 και στο Σχήμα 5.7 του αλγορίθμου YOLOv5, μελετήθηκε ένα σύνολο δεδομένων 500 εικόνων ομοειδών αντικειμένων που περιείχε το Αντικείμενο-2 . Κάνοντας μια σύγκριση των αποτελεσμάτων με αυτά που προκύπτουν από το σύνολο δεδομένων



Σχήμα 5.7: Μέτρηση αλγορίθμου YOLO

του Αντικειμένου-3, συμπεραίνουμε ότι η απόδοση του αλγορίθμου υστερεί στο σύνολο δεδομένων Αντικειμένου-2 για την μετρική του Accuracy σε μεγάλο βαθμό και στις μετρικές του F1-score και του Recall σε μικρότερο. Το Precision είναι σχεδόν το ίδιο όπως επίσης και ο χρόνος εκτέλεσης του αλγορίθμου και στα δύο σύνολα δεδομένων.

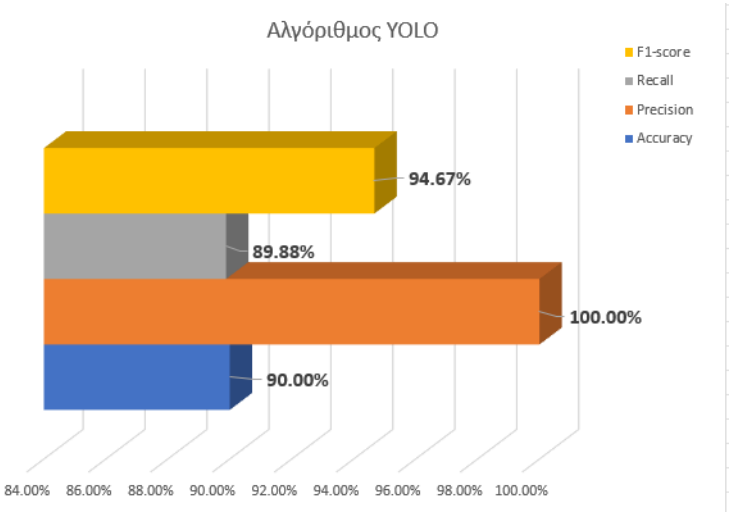
Πίνακας 5.7: Μετρικές αλγορίθμου YOLO.

Μετρική	**Τιμή**
Accuracy	86.03
Precision	100.00
Recall	85.14
F1-score	91.97

Χρόνος εκτέλεσης : 0.31 λεπτά

- Μοντέλο: YOLOv5
- Dataset: 500 εικόνες του ίδιου Αντικειμένου-3 (ελέφαντας)

Ο Πίνακας 5.8 και το Σχήμα 5.8 περιέχουν τα αποτελέσματα που προκύπτουν από την μέτρηση του Accuracy, του Recall, του Precision και του F1-score για τον αλγόριθμο YOLOv5, εκτελεσμένο σε ένα σύνολο δεδομένων που περιείχε μόνο το Αντικείμενο-3 και είχε μέγεθος 500 εικόνες. Παρατηρούμε ότι κάνοντας χρήση αυτού του συνόλου δεδομένων, επιτυγχάνεται η καλύτερη απόδοση σε σχέση με αυτές



Σχήμα 5.8: Μέτρηση αλγορίθμου YOLO

που προέκυψαν στην μελέτη για την υλοποίηση των υπολοίπων δύο συνόλων δεδομένων, που το καθένα περιείχε μόνο Αντικείμενο-1 ή Αντικείμενο-2 αντίστοιχα. Οι μόνες μετρήσεις οι οποίες έχουν μικρή απόκλιση από σύνολο δεδομένων σε σύνολο δεδομένων είναι αυτή του Precision και του χρόνου.

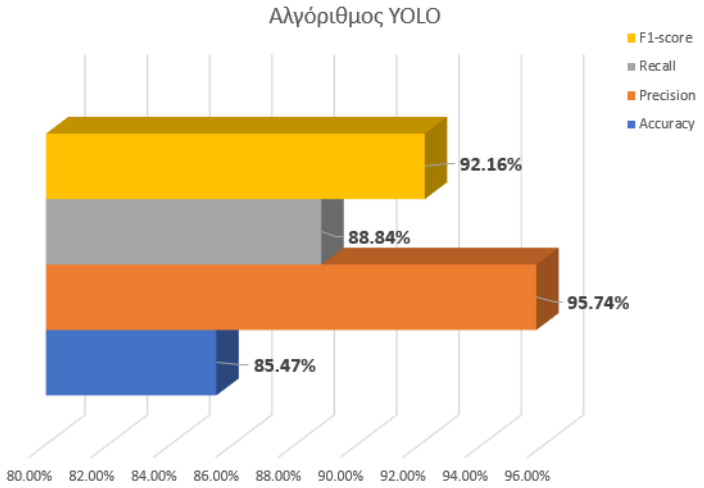
Πίνακας 5.8: Μετρικές αλγορίθμου YOLO.

Μετρική	**Τιμή**
Accuracy	90.00
Precision	100.00
Recall	89.88
F1-score	94.67

Χρόνος εκτέλεσης : 0.30 λεπτά

- Μοντέλο: YOLOv5
- Dataset: 1,500 εικόνες Αντικείμενο-1 500, Αντικείμενο-2 500, Αντικείμενοο-3 500.

Στον Πίνακα 5.9 και Σχήμα 5.9 εμπεριέχονται οι μετρήσεις του αλγορίθμου YOLOv5 σε σύνολο δεδομένων 1,500 εικόνων, που προκύπτει από την συνένωση των τριών συνόλων δεδομένων που περιέχουν το Αντικείμενο-1, το Αντικείμενο-2 και το Αντικείμενο-3 αντίστοιχα. Όπως μπορεί να γίνει αντιληπτό μετά από μια σύντομη σύγκριση των αποτελεσμάτων με αυτά του συνόλου δεδομένων που περιέχει μόνο το Αντικείμενο-2, οι μετρήσεις του F1-score και του Recall αυξάνονται λίγο σε αυτή την συνθήκη, ενώ οι μετρήσεις του Accuracy και του Precision μειώνονται λίγο. Ο



Σχήμα 5.9: Μέτρηση αλγορίθμου YOLO

χρόνος όπως είναι αναμενόμενο αυξάνεται αφού το μέγεθος του συνόλου δεδομένων μεγαλώνει.

Πίνακας 5.9: Μετρικές αλγορίθμου YOLO.

Μετρική	**Τιμή**
Accuracy	85.47
Precision	95.74
Recall	88.84
F1-score	92.16

Χρόνος εκτέλεσης: 0.77 λεπτά

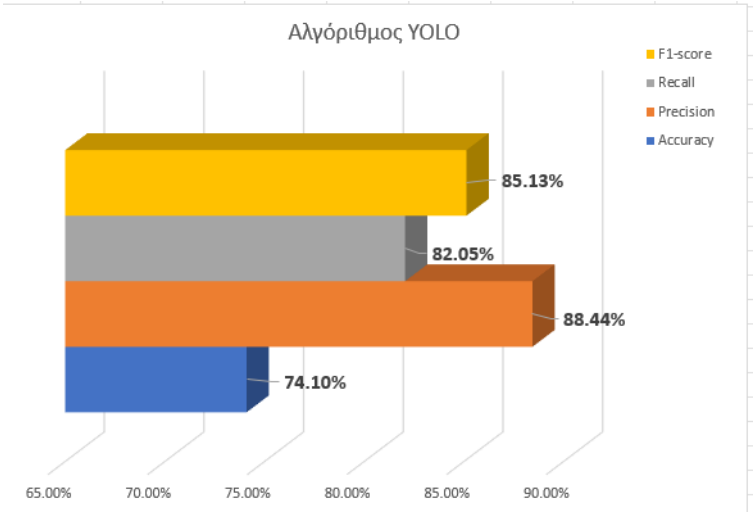
- Μοντέλο: YOLOv5
- Dataset: 3,000 εικόνες Αντικείμενο-1 500, Αντικείμενο-2 500, Αντικείμενο-3 500 και 1,500 εικόνες άλλον αντικειμένων (διαφορετικά ζώα).

Στον Πίνακα 5.10 και στο Σχήμα 5.10 αποτυπώνονται τιμές του αλγορίθμου YOLOv5 για ένα σύνολο δεδομένων 3,000 εικόνων που αποτελείται από 1,500 εικόνες των προς ανίχνευση αντικειμένων 1,2,3 και 1,500 εικόνες διαφορετικών αντικειμένων. Μετά από κατάλληλη παρατήρηση με το σύνολο δεδομένων των 1,500 εικόνων προκύπτει ότι οι τιμές των Accuracy, Precision, Recall και F1-score αυξάνονται αρκετά, οπότε η αύξηση του μεγέθους του συνόλου δεδομένων έχει θετική επίδραση στα αποτελέσματα του αλγορίθμου.

Χρόνος εκτέλεσης: 1.48 λεπτά

Πίνακας 5.10: Μετρικές αλγορίθμου YOLO.

Μετρική	Τιμή
Accurancty	96.00
Precision	99.53
Recall	96.00
F1-score	98.00



Σχήμα 5.10: Μέτρηση αλγορίθμου YOLO

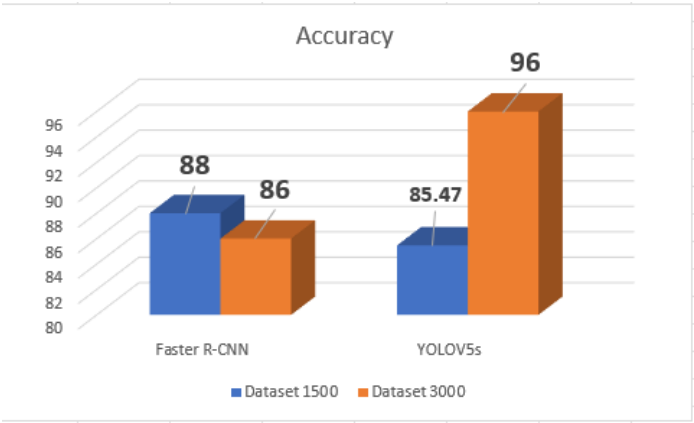
Η επιλογή των δυο εικόνων που παρουσιάζονται παρακάτω, Σχήματα 5.11 και 5.12, έγινε σκόπιμα για να εξεταστεί η ακρίβεια του αλγορίθμου. Κατά την εκτέλεση του αλγορίθμου διαπιστώθηκε η λανθασμένη αναγνώριση του αντικειμένου πεταλούδα στο Σχήμα 5.11 και η τοποθέτησή της στην κατηγορία του αντικειμένου-3 (ελέφαντας), λόγω της ομοιότητας και του μεγάλου όγκου που καταλαμβάνουν τα χαρακτηριστικά των φτερών της πεταλούδας σαν αυτιά του ελέφαντα (Σχήμα 5.12). Παρά όλες τις προσπάθειες που έγιναν για τη βελτιστοποίηση της συμπεριφοράς και την αποφυγή του συγκεκριμένου αποτελέσματος αυτό δεν κατέστη δυνατό.

Η σύγκριση έγινε για dataset 1,500 και 3,000 εικόνων και φαίνεται στο Σχήμα 5.13. Το Accuracy του αλγορίθμου YOLO αυξάνεται σημαντικά με την αύξηση του μεγέθους του dataset. Η τιμή αυτή μειώνεται στον αλγόριθμο Faster R-CNN ανάλογα με το μέγεθος του dataset όχι όμως σε τόσο μεγάλο βαθμό.

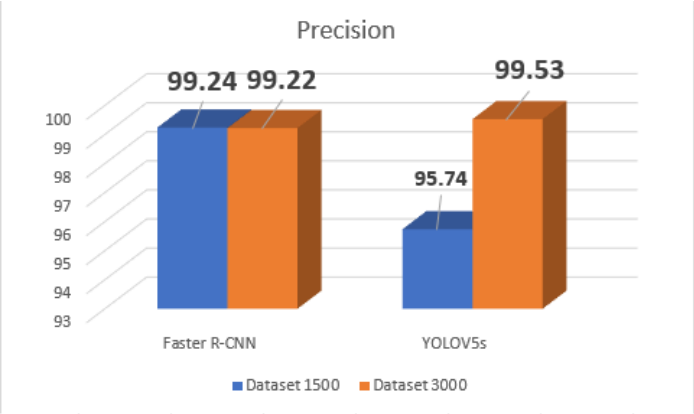
Με σκοπό να υπάρξουν οι επιθυμητές μετρήσεις χρησιμοποιήθηκαν δύο dataset 1,500 και 3,000 εικόνων και παρουσιάζονται στο Σχήμα 5.14. Η τιμή του Precision για τον αλγορίθμου Faster R-CNN είναι ανεξάρτητη από το μέγεθος του dataset. Ο αλγόριθμος YOLO πολύ καλύτερη απόδοση όσο αυξάνεται το μέγεθος του dataset.

Σχήμα 5.11: Πεταλούδα

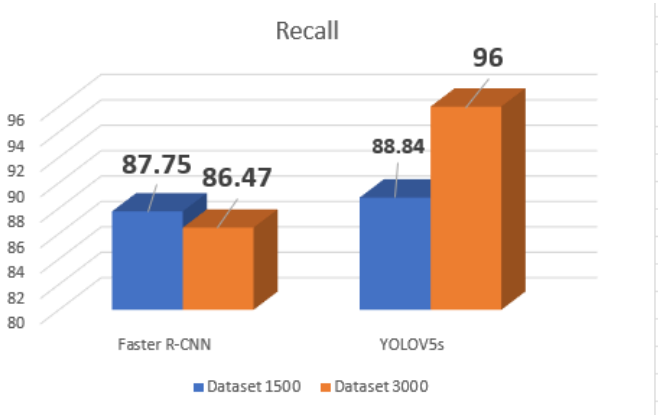
Σχήμα 5.12: Ελέφαντας με αυτιά πεταλούδας



Σχήμα 5.13: Σύγκριση Accuracy αλγορίθμων Faster R-CNN και YOLO



Σχήμα 5.14: Σύγκριση Precision αλγορίθμων Faster R-CNN και YOLO

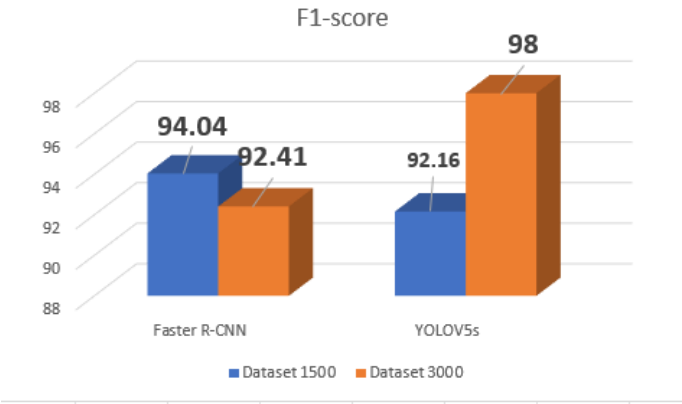


Σχήμα 5.15: Σύγκριση Recall αλγορίθμων Faster R-CNN και YOLO

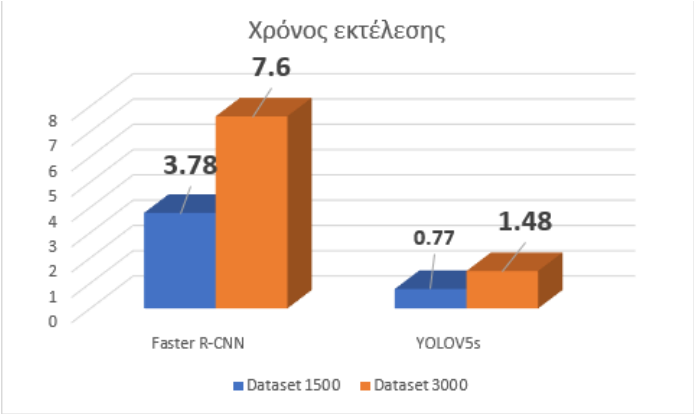
Για να προκύψει η σύγκριση των δύο αλγορίθμων που απεικονίζεται στο Σχήμα 5.15 έγινε χρήση dataset 1,500 και 3,000 εικόνων. Ο αλγόριθμος YOLO παρουσιάζει πολύ καλύτερο Recall όταν το dataset των εικόνων αυξάνεται σε μέγεθος. Η αύξησή του είναι ανάλογη της αύξησης του μεγέθους του dataset. Ο Faster R-CNN λειτουργεί αντίστροφα καθώς παρατηρούμε ότι η αύξηση του μεγέθους του dataset έχει ως αποτέλεσμα την μείωση της συγκεκριμένης τιμής.

Οι μετρήσεις για την αξιολόγηση της μετρικής F1-score που αναπαριστώνται στο Σχήμα 5.16 έχουν δημιουργηθεί εκτελώντας τους αλγόριθμους για dataset 1,500 και 3,000 εικόνων. Το F1-score του αλγορίθμου YOLO αυξάνεται αναλογικά σε σχέση με την αύξηση του μεγέθους του dataset. Σημαντική είναι η μείωση της συγκεκριμένης τιμής με την χρήση μεγαλύτερου μεγέθους dataset για τον αλγόριθμο Faster R-CNN.

Ο χρόνος εκτέλεσης του αλγορίθμου Faster R-CNN αυξάνεται με την χρήση μεγαλύτερου dataset εικόνων (από 1,500 σε 3,000 εικόνες) και είναι αναλογική. Ο αλγόριθμος YOLO παρουσιάζει την ίδια αναλογική αύξηση για την ίδια μεταβολή του dataset. Οι χρόνοι εκτέλεσης του αλγορίθμου YOLO εξαιρετικά καλύτεροι σε



Σχήμα 5.16: Σύγκριση F1-score αλγορίθμων Faster R-CNN και YOLO



Σχήμα 5.17: Σύγκριση χρόνου εκτέλεσης αλγορίθμων Faster R-CNN και YOLO

σχέση με του Faster R-CNN και παρατηρείται ότι συνδέονται μεταξύ τους με μια αναλογία 1/5.

Από την υπολογιστική μελέτη και την εξέταση των συγκριτικών διαγραμμάτων προκύπτει πως ο αλγόριθμος αναγνώρισης αντικειμένων YOLO υπερτερεί σημαντικά σε όλες τις μετρικές. Η απόδοσή του αυξάνεται με την αύξηση του μεγέθους του dataset των εικόνων ενώ η απόδοση του Faster R-CNN είναι αντιστρόφως ανάλογη. Σημαντικός παράγοντας είναι και ο χρόνος εκτέλεσης των δύο αλγορίθμων. Ο YOLO είναι εντυπωσιακά πιο γρήγορος συγκρινόμενος με το ίδιο μέγεθος δεδομένων.

5.2 Αλγόριθμοι Κρυπτογράφησης εικόνων

Η μελέτη των αλγορίθμων κρυπτογράφησης και η σύγκρισή τους έγινε σε σύνολο δεδομένων 500 εικόνων.

- Αλγόριθμος: DES

- Dataset: 500 εικόνες

Πίνακας 5.11: Μετρικές αλγορίθμου DES.

Μετρική	**Τιμή**
Μέγεθος κλειδιού	128,192,256 bit
Είδος κρυπτογράφησης	Συμμετρική
Γύροι για ολοκλήρωση κρυπτογράφησης	16
Χρόνος που απαιτείται για την κρυπτογράφηση	9.54 s
Χρόνος που απαιτείται για την αποκρυπτογράφηση	1.50 s

- Αλγόριθμος: AES

- Dataset: 500 εικόνες

Πίνακας 5.12: Μετρικές αλγορίθμου AES.

Μετρική	**Τιμή**
Μέγεθος κλειδιού	128,192,256 bit
Είδος κρυπτογράφησης	Συμμετρική
Γύροι για ολοκλήρωση κρυπτογράφησης	10/128-bit,12/192-bit,14/256-bit
Χρόνος που απαιτείται για την κρυπτογράφηση	1.20 s
Χρόνος που απαιτείται για την αποκρυπτογράφηση	0.25 s

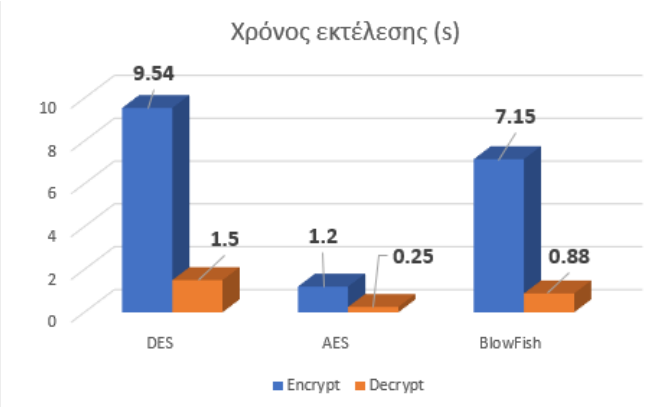
- Αλγόριθμος: BlowFish

- Dataset: 500 εικόνες

Πίνακας 5.13: Μετρικές αλγορίθμου BlowFish.

Μετρική	**Τιμή**
Μέγεθος κλειδιού	32-448 bit
Είδος κρυπτογράφησης	Συμμετρική
Γύροι για ολοκλήρωση κρυπτογράφησης	16
Χρόνος που απαιτείται για την κρυπτογράφηση	7.15 s
Χρόνος που απαιτείται για την αποκρυπτογράφηση	0.88 s

Από τους Πίνακες 5.11, 5.12 και 5.13 και από το Σχήμα 5.18 συμπεραίνουμε ότι ο χρόνος που απαιτείται για την κρυπτογράφηση είναι μεγαλύτερος από αυτόν που χρειάζονται τα δεδομένα για να αποκρυπτογραφηθούν και αυτό ισχύει και για τους τρεις αλγορίθμους DES, AES, BlowFish. Ο αλγόριθμος AES είναι εξαιρετικά πιο γρήγορος, ακολουθεί ο αλγόριθμος BlowFish και τελευταίος είναι ο αλγόριθμος



Σχήμα 5.18: Σύγκριση χρόνου εκτέλεσης αλγορίθμων DES, AES, BlowFish

DES. Η διαφορά αυτή στο χρόνο παρατηρείται γιατί ο αλγόριθμος AES χρειάζεται λιγότερους γύρους για να ολοκληρώσει την κρυπτογράφηση, ανεξάρτητα από το μήκος κλειδιού που χρησιμοποιεί σε σχέση με το DES και το BlowFish. Οι υπόλοιποι δύο αλγόριθμοι έχουν τον ίδιο αριθμό γύρων για την ολοκλήρωση της κρυπτογράφησης.

Όσον αφορά το μέγεθος του κλειδιού που χρησιμοποιεί κάθε αλγόριθμος για κρυπτογράφηση και αποκρυπτογράφηση, οι αλγόριθμοι AES και DES μπορούν να χρησιμοποιούν το ίδιο μέγεθος κλειδιού 128,192,256 bit ενώ ο αλγόριθμος BlowFish χρησιμοποιεί κλειδί μήκους 32-448 bit. .

Όλα αυτά συμβάλουν στην διαπίστωση ότι ο αλγόριθμος AES είναι πιο αποδοτικός από τους άλλους δύο αλγόριθμους κρυπτογράφησης που μελετήσαμε.

Κεφάλαιο 6

Συμπεράσματα

Στη συγκεκριμένη διπλωματική εργασία μελετήθηκε ο τρόπος εντοπισμού και αναγνώρισης αντικειμένων σε εικόνες και βίντεο χρησιμοποιώντας αλγόριθμους Νευρωνικών Δικτύων. Επίσης έγινε έρευνα για τους αλγόριθμους κρυπτογράφησης-αποκρυπτογράφησης που μπορούν να χρησιμοποιηθούν, για να επιτευχθεί η διαφύλαξη του περιεχομένου αυτών από πρόσβαση σε τρίτους, μη εξουσιοδοτημένους αναγνώστες.

Αρχικά έγινε μια μελέτη των διαθέσιμων αλγορίθμων αναγνώρισης αντικειμένων και στη συνέχεια ακολούθησε μια σύγκριση και υλοποίηση δύο αλγορίθμων από αυτούς, του YOLO και του Faster R-CNN. Ο αλγόριθμος Faster R-CNN ανήκει στην κατηγορία αλγορίθμων βαθύ συνελικτικού δικτύου, ενώ ο αλγόριθμος YOLO στην κατηγορία αλγορίθμων ανίχνευσης ενός σταδίου. Πραγματοποιώντας την υπολογιστική μελέτη και την σύγκριση των δύο αυτών αλγορίθμων σε σύνολα δεδομένων 1,500 και 3,000 φωτογραφιών, μπορέσαμε να καταλήξουμε σε κάποια συμπεράσματα. Ο αλγόριθμος YOLO είναι πιο ακριβής, δηλαδή έχει μεγαλύτερο ποσοστό για τη μετρική της Accuracy με ποσοστά 85.47 και 96 τις εκατό, για τα σύνολα δεδομένων μεγέθους 1,500 και 3,000 έναντι του αλγορίθμου Faster R-CNN με ποσοστά 88 και 86 της εκατό αντίστοιχα. Αυτή η υπεροχή του αλγορίθμου YOLO συγκριτικά με τον αλγόριθμο Faster R-CNN αντικατοπτρίζεται και στις υπόλοιπες μετρήσεις που αφορούν τις μετρικές Recall, Precision και F1-score. Επιπλέον έγινε σύγκριση του χρόνου που χρειάζεται για να πραγματοποιηθεί η αναγνώριση αντικειμένων και για τους δύο αλγόριθμους σε ίσου μεγέθους σύνολα δεδομένων. Ο YOLO είναι πιο γρήγορος με χρόνο 0.77 και 1.48 για σύνολα δεδομένων 1,500 και 3,000 εικόνων, ενώ ο Faster R-CNN χρειάστηκε χρόνο 3.78 και 7.6 αντίστοιχα. Τέλος ένας πολύ σημα-

ντικός παράγοντας για την επίτευξη της καλύτερης απόδοσης των δύο αλγορίθμων είναι και η επιλογή του κατάλληλου μοντέλου. Για τον αλγόριθμο YOLO χρησιμοποιήθηκε το μοντέλο YOLOv5, ενώ για τον Faster R-CNN το μοντέλο ResNet50. Όλα αυτά συμβάλλουν στην εξαγωγή του συμπεράσματος ότι ο αλγόριθμος YOLO είναι ο κατάλληλος για την αναγνώριση των αντικειμένων μας.

Εν συνεχεία μελετήθηκαν οι αλγόριθμοι κρυπτογράφησης και αποκρυπτογράφησης που μπορούν να χρησιμοποιηθούν σε εικόνες. Αρχικά έγινε ανάλυση των δύο κατηγοριών αλγορίθμων, συμμετρικής και ασύμμετρης κρυπτογράφησης και διαπιστώθηκε ότι οι αλγόριθμοι συμμετρικής κρυπτογράφησης είναι αυτοί που μπορούν να πραγματοποιήσουν ορθότερα την κρυπτογράφηση και αποκρυπτογράφηση εικόνων. Στην συνέχεια υπήρξε κατάλληλη υπολογιστική μελέτη και σύγκριση των αλγορίθμων DES, AES και BlowFish. Οι αλγόριθμοι αυτοί συγκρίθηκαν ως προς τις μετρικές : μέγεθος κλειδιού, γύροι για ολοκλήρωση κρυπτογράφησης, χρόνος κρυπτογράφησης και χρόνος αποκρυπτογράφησης σε ένα σύνολο δεδομένων 500 εικόνων. Έτσι καταλήξαμε στα συμπεράσματα ότι ο αλγόριθμος AES είναι πιο αποδοτικός, καθώς πραγματοποιεί την κρυπτογράφηση και την αποκρυπτογράφηση σε μικρότερο χρονικό διάστημα σε σχέση με τους αλγόριθμους DES και BlowFish. Πραγματοποιεί την κρυπτογράφηση σε λιγότερους γύρους σε σχέση με τους υπόλοιπους 10 ή 12 ή 14 γύρους, ανάλογα με το μέγεθος κλειδιού που χρησιμοποιείται έναντι των 16 γύρων των υπόλοιπων δυο αλγορίθμων. Άρα γίνεται κατανοητό ότι για τέτοιου είδους εφαρμογές ο αλγόριθμος AES είναι πιο κατάλληλος.

6.1 Μελλοντικές επεκτάσεις

Η ενότητα αυτή περιέχει εφαρμογές στις οποίες μπορεί να χρησιμοποιηθεί η αναγνώριση αντικειμένων σε εικόνες και βίντεο καθώς επίσης και η κρυπτογράφηση σε αυτά, με σκοπό τη διαφύλαξη των δεδομένων προσωπικού χαρακτήρα στην εκάστοτε περίπτωση.

Τομέας της υγείας

Οι αλγόριθμοι αναγνώρισης αντικειμένων μπορούν να χρησιμοποιηθούν στην ανίχνευση ασθενειών από φωτογραφίες μαγνητικών και αξονικών τομογράφων με αποτέλεσμα να πραγματοποιείται πιο ακριβής και έγκυρη διάγνωση αυτών. Μέσω αυτής της εφαρμογής οι επιστήμονες που εργάζονται σε τέτοιου είδους εργαστήρια,

αλλά και οι ιατροί έχουν την δυνατότητα να κάνουν διαγνώσεις με μεγαλύτερη ευκολία και να αντιμετωπίζουν το πρόβλημα πιο άμεσα. Επιπλέον η αναγνώριση αντικειμένων μπορεί να έχει εφαρμογή στην χειρουργική, καθώς μπορούμε με τη χρήση κατάλληλων αλγορίθμων να γίνει εντοπισμός των διάφορων ζωτικών οργάνων, ανεξάρτητα από την ξεχωριστή μορφολογία κάθε οργανισμού και να είναι πιο στοχευμένη η επέμβαση σε αυτά. Πέρα από την αναγνώριση αντικειμένων και οι αλγόριθμοι κρυπτογράφησης βρίσκουν εφαρμογή στον τομέα αυτό, διότι χρησιμοποιούνται για την ασφαλή διακίνηση των τόσο ευαίσθητων δεδομένων μεταξύ των επιστημόνων. Έτσι οι εικόνες κρυπτογραφούνται, στέλνονται προς εξέταση και στην συνέχεια μόλις φτάσουν στον παραλήπτη αποκρυπτογραφούνται.

Επιτήρηση ασφαλείας

Οι αλγόριθμοι αναγνώρισης αντικειμένων μπορούν να χρησιμοποιηθούν σε συστήματα παρακολούθησης για λόγους ασφαλείας. Δηλαδή οι αλγόριθμοι αυτοί εντοπίζουν την δραστηριότητα αντικειμένων σε πραγματικό χρόνο, και στην συνέχεια την κατατάσσουν σε φυσιολογική, ύποπτη ή επικίνδυνη. Στην συνέχεια γίνονται οι κατάλληλες ενέργειες ειδοποιώντας τους υπεύθυνους για την αντιμετώπιση κάθε κατάστασης. Και σε αυτή την εφαρμογή η κρυπτογράφηση των εικόνων θεωρείται ζωτικής σημασίας γιατί τα δεδομένα κατατάσσονται στην κατηγορία των ευαίσθητων και η διαρροή τους σε τρίτους προσβάλει την ιδιωτικότητα των ατόμων που απεικονίζονται στις υπό επεξεργασία εικόνες ή βίντεο.

Αυτοκινούμενα οχήματα

Στον τομέα αυτό χρησιμοποιούνται οι αλγόριθμοι αναγνώρισης αντικειμένων έτσι ώστε να γίνεται αντιληπτό από τα οχήματα, χωρίς την ανθρώπινη παρέμβαση, τι υπάρχει γύρω τους. Επομένως μπορούν να αναγνωρίζουν την ύπαρξη πεζών, ζώων, πινακίδων, οχημάτων, φαναριών και να πραγματοποιούν την ανάλογη πράξη. Με αυτόν τον τρόπο εξασφαλίζεται η ασφάλεια των επιβατών τους, αλλά και των γύρω αντικειμένων. Και σε αυτή την εφαρμογή επιβάλλεται η διαφύλαξη των προσωπικών δεδομένων όπως είναι τα πρόσωπα και οι αριθμοί πινακίδων των υπολοίπων οχημάτων. Αυτό επιτυγχάνεται από την κρυπτογράφηση των εικόνων με την χρήση κατάλληλων αλγορίθμων.

Βιβλιογραφία

[1] Ifeoluwapo Aribilola, Mamoona Naveed Asghar, Nadia Kanwal, Martin Fleury, and Brian Lee. Securecam: Selective detection and encryption enabled application for dynamic camera surveillance videos. *IEEE Transactions on Consumer Electronics*, 69(2):156–169, 2023.

[2] Sebastian Castillo, Anyela Bernal, and Jorge Rodríguez. Object detection in digital documents based on machine learning algorithms. *IAENG International Journal of Computer Science*, 50(2), 2023.

[3] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. *Signal processing*, 90(3):727–752, 2010.

[4] Min-Kyu;Lee Cheon. Rotation invariant histogram of oriented gradients.

[5] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in neural information processing systems*, 29, 2016.

[6] ArcGIS Develope. How single-shot detector (ssd) works?, 2022.

[7] Pengfei Fang, Han Liu, Chengmao Wu, and Min Liu. A survey of image encryption algorithms based on chaotic system. *The Visual Computer*, 39(5):1975–2003, 2023.

[8] Alvaro Fuentes, Sook Yoon, Sang Cheol Kim, and Dong Sun Park. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9):2022, 2017.

[9] Ahmed Fawzy Gad. Faster r-cnn explained for object detection tasks. *URL: https://blog. paperspace. com/faster-r-cnn-explained-object-detection/.(zugegriffen: 04.07. 2022)*, 2021.

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[12] Larry Hardesty. Explained: Neural networks, Apr 2017.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *Computer Vision – ECCV 2014*, pages 346–361. Springer International Publishing, 2014.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[15] Jonathan Hui. Ssd object detection: Single shot multibox detector for real-time processing. *Jonathan Hui.–URL: https://jonathan-hui. medium. com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06 (date of application: 10.10. 2021)*, 2018.

[16] Omar A Imran, Sura F Yousif, Isam Salah Hameed, Wisam Najm Al-Din Abed, and Ali Thaeer Hammid. Implementation of el-gamal algorithm for speech signals encryption and decryption. *Procedia Computer Science*, 167:1028–1037, 2020.

[17] Harshil Jain and SK Nandy. Incremental training for image classification of unseen objects. *ResearchGate, Aug*, 2019.

[18] AL Jeeva, Dr V Palanisamy, and K Kanagaram. Comparative analysis of performance efficiency and security measures of some encryption algorithms. *International Journal of Engineering Research and Applications (IJERA)*, 2(3):3033–3037, 2012.

[19] Huaizu Jiang and Erik Learned-Miller. Face detection with the faster r-cnn. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pages 650–657. IEEE, 2017.

[20] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022.

[21] Jianwei Li, Changwen Qu, and Jiaqi Shao. Ship detection in sar images based on an improved faster r-cnn. In *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, pages 1–6. IEEE, 2017.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing, 2016.

[23] Yang Liu, Zhuo Ma, Ximeng Liu, Siqi Ma, and Kui Ren. Privacy-preserving object detection for medical images with faster r-cnn. *IEEE Transactions on Information Forensics and Security*, 17:69–84, 2019.

[24] Faiqa Maqsood, Muhammad Ahmed, Muhammad Mumtaz Ali, and Munam Ali Shah. Cryptography: A comparative analysis for modern techniques. *International Journal of Advanced Computer Science and Applications*, 8(6), 2017.

[25] Evgeny Milanov. The rsa algorithm. *RSA laboratories*, pages 1–11, 2009.

[26] Shubh Mody, Hriday Mehta, Pragun Mantri, Bushra Ali, and Ankit Khivasara. Safety gear equipment detection for warehouse and construction sites using yolov5. 2021.

[27] Komal D Patel and Sonal Belani. Image encryption using different techniques: A review. *International Journal of Emerging Technology and Advanced Engineering*, 1(1):30–34, 2011.

[28] Priyanka and Amit Kumar Singh. A survey of image encryption for healthcare applications. *Evolutionary Intelligence*, 16(3):801–818, 2023.

[29] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[30] Vincent Rijmen and Joan Daemen. Advanced encryption standard. *Proceedings of federal information processing standards publications, national institute of standards and technology*, 19:22, 2001.

[31] Aradhana Sahoo, Pratyasha Mohanty, and Purna Chandra Sethi. Image encryption using rsa algorithm. In *Intelligent Systems: Proceedings of ICMIB 2021*, pages 641–652. Springer, 2022.

[32] Bruce Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *International Workshop on Fast Software Encryption*, pages 191–204. Springer, 1993.

[33] Prabhdeep Singh, Vikas Tripathi, Durgaprasad Gangodkar, and Dibyahash Bordoloi. A des, aes, dss, and rsa-based security system for protecting sensitive information during communication and providing fast, reliable file identification. *Webology*, 18(5):3218–3227, 2021.

[34] Shrey Srivastava, Amit Vishvas Divekar, Chandu Anilkumar, Ishika Naik, Ved Kulkarni, and V Pattabiraman. Comparative analysis of deep learning image detection algorithms. *Journal of Big data*, 8(1):1–27, 2021.

[35] Jawahar Thakur and Nagesh Kumar. Des, aes and blowfish: Symmetric key cryptography algorithms simulation based performance analysis. *International journal of emerging technology and advanced engineering*, 1(2):6–12, 2011.

[36] Sun-Chong Wang and Sun-Chong Wang. Artificial neural network. *Interdisciplinary computing in java programming*, pages 81–100, 2003.

[37] Xingyuan Wang and Yafei Wang. Multiple medical image encryption algorithm based on scrambling of region of interest and diffusion of odd-even interleaved points. *Expert Systems with Applications*, 213:118924, 2023.

[38] Sura F. Yousif, Ali J. Abboud, and Hussein Y. Radhi. Robust image encryption with scanning technology, the el-gamal algorithm and chaos theory. *IEEE Access*, 8:155184–155209, 2020.

[39] Chunyang Zhu, Weihua Zhao, and Heng Lian. Image recognition and classification with hog based on nonlinear support tensor machine. *Multimedia Tools and Applications*, pages 1–20, 2022.

[40] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *Proceedings of the IEEE*, 2023.