ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΠΡΟΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

Brain tumor detection using Faster R-CNN detector

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της/του

Φουράκης Δημήτριος

Επιβλέπων :Καβαλλιεράτου Εργίνα
Βασιλόπουλος Νικόλαος

Μέλη εξεταστικής επιτροπής: Καβαλλιεράτου Εργίνα
Βασιλόπουλος Νικόλαος
Σταματάτος Ευστάθιος

Σάμος, Φεβρουάριος 2020

Πρόλογος και ευχαριστίες

Θα ήθελα να ευχαριστήσω την καθηγήτρια μου κυρία Εργίνα Καβαλιεράτου, για την εμπιστοσύνη που μου έδειξε με την ανάθεση της συγκεκριμένης διπλωματικής εργασίας και τον επιβλέποντα, κύριο Νικόλαο Βασιλόπουλο, για τις συμβουλές του σε όλη την πορεία της συγγραφής της εργασίας αυτής.

Φουράκης Δημήτριος
Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων
ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Πίνακας περιεχομένων

Contents

Λίστα Σχημάτων

Ακρωνύμια

AI	Artificial Intelligence
ANN	Artificial Neural Networks
API	Application Programming Interface
CNN	Convolutional Neural Network
CPU	Computer Processing
DAG	Directed Acyclic Graph
GPU	Graphic Processing
ML	Machine Learning
NN	Neural Network
RPN	Region Proposal Network
SVM	Support Vector Machine
TPU	Tensor Processing Unit
LGG	Low Grade Glioma
HGG	High Grade Glioma

Περίληψη

Σε αυτή την εργασία θα ασχοληθούμε με τις λειτουργίες ενός συστήματος Υπολογιστικής όρασης και πιο συγκεκριμένα, ενός αυτοματοποιημένου συστήματος που έχει ως σκοπό το object recognition, και detection, και στο δεύτερο μέρος της εργασίας γίνεται υλοποίηση μίας προσέγγισης συνελικτικών νευρωνικών δικτύων για την ανίχνευση όγκου στον εγκέφαλο.

Αρχικά θα αποσαφηνιστούν κάποιες βασικές αρχές, οι οποίες σχετίζονται με τις παραπάνω λειτουργίες. Αυτές οι αρχές αφορούν την ταξινόμηση, τον ταξινομητής αντικειμένου, την ανίχνευση αντικειμένου, την επεξεργασία των εικόνων, τη συλλογή δεδομένων και τη δημιουργία ενός μοντέλου το οποίο συνδυάζει τα χρήσιμα χαρακτηριστικά που εξάγονται με κάποιους αλγορίθμους από κάποιο dataset. Στόχος είναι η χρήση των παραπάνω στην εξαγωγή αποφάσεων, αναγνώρισης αντικειμένων.

Στη συνέχεια θα αναφερθούμε στους παράγοντες που δυσκολεύουν την αναγνώριση αντικειμένου αλλά και στις προσεγγίσεις αναγνώρισης του αντικειμένου. Κατόπιν γίνεται αναφορά στους ανιχνευτές υψηλής τεχνολογίας CNN (Faster R-CNN, YOLO, SSD) και ειδικότερα στο μοντέλο Faster R-CNN. Τέλος γίνεται αναφορά στα εργαλεία που χρησιμοποιήθηκαν (python, OpenCV, το Tensorflow, το CUDA και την βιβλιοθήκη Nvidia Cudnn).

Για την υλοποίηση του δεύτερου μέρους θα εκπαιδευτεί ένας ταξινομητής ανίχνευσης αντικειμένων σε συνελικτίκο νευρωνικό δίκτυο, με σκοπό την αναγνώριση όγκου στον εγκέφαλο. Το εργαλείο που χρησιμοποιείται είναι το Tensorflow-GPU 1.8 CUDA v10.0 και το cuDNN 7.4, που είναι συμβατά με το tensorflow-gpu Python 3.6 και το Anaconda environment. Στην συνέχεια θα αναφερθούν συνοπτικά τα βήματα της υλοποίησης. Κατόπιν, περιγράφονται αναλυτικά τα βήματα που ακολουθήθηκαν ενώ παρατίθενται και εικόνες επεξήγησης του κάθε βήματος.

Συμπερασματικά, στην παρούσα εργασία περιγράψαμε διάφορους αλγόριθμους οπτικής ανίχνευσης αντικειμένων συμπεριλαμβανομένων μερικών από τους state-of-the-art ανιχνευτών CNN που θα μπορούσαν να χρησιμοποιηθούν για ανίχνευση όγκου στον εγκέφαλο. Από αυτούς τους ανιχνευτές, χρησιμοποιήσαμε το faster r-cnn σύστημα, το οποίο αποδείχθηκε αποτελεσματικό και εύκολο να τροποποιηθεί για εφαρμογή. Παράλληλα, δίνονται και κατευθύνσεις για μελλοντικές βελτιώσεις για την συγκεκριμένη υλοποίηση.

Λέξεις Κλειδιά: *CNN, Faster R-CNN, state of the art*

Abstract

In this thesis we will deal with the functions of a Computer Vision system and more specifically with a automated system for object recognition, detection and tracking, while in the second part of the thesis we implement an approach of convolutional neural networks for the detection of brain tumor.

Initially, some basic principles related to the above functions will be clarified. These principles relate to classification, object classifier, object detection, image processing, object detection, data set, models that combines useful features extracted with some algorithms from a dataset for their use export of decisions, recognition of objects.

We will then refer to the factors that make it difficult to identify an object but also to the object recognition approaches. Reference is then made state of the art CNN detectors (Faster R-CNN, YOLO, SSD), and particular to the faster R-CNN models. Finally, reference is made to the tools used (python, openCV, Tensorflow, CUDA and the Nvidia Cudnn library).

For the implementation of the second part an object detector classifier will be trained on a neuronal network to identify brain tumor using Tensorflow-GPU 1.8 CUDA v10.0 and cuDNN 7.4 that are compatible with tensorflow-gpu Python 3.6 and Anaconda environment. Below is a brief summary of the implementation steps. In the second part, the following steps are described in detail, and illustrative illustrations of each step are presented.

In conclusion, we have described various optical object detection algorithms, including some of the state-of-the-art CNN detectors that could be used to brain tumor detection. From these detectors, we used the faster r-cnn system, which proved to be effective and easy to modify for application. At the same time, directions are given for future improvements for this implementation.

Keywords: *CNN, Faster R-CNN, state of the art*

1 *Εισαγωγή*

1.1 Brain tumor detection using convolutional neural network

Σε αυτή την εργασία θα ασχοληθούμε με τις λειτουργίες ενός συστήματος Υπολογιστικής όρασης και πιο συγκεκριμένα ενός συστήματος που έχει ως σκοπό το object recognition και detection . Στο πρώτο μέρος θα αναλυθούν οι αλγόριθμοι αναγνώρισης αντικειμένου. Κατόπιν θα δημιουργήσουμε ένα μοντέλο για την εξαγωγή της απόφασης, το οποίο στην ουσία είναι ένα instance ενός Νευρωνικού Δικτύου που έχει δημιουργηθεί με κάποια από της μεθόδους δημιουργίας νευρωνικών δικτύων. Πιο συγκεκριμένα, η υλοποίηση θα αφορά ένα αυτοματοποιημένο σύστημα ανίχνευσης όγκου στον εγκέφαλο με βάση τα δεδομένα εκπαίδευσης, τα οποία είναι ένα σύνολο μαγνητικών τομογραφιών. Η εκπαίδευση θα γίνει με τη χρήση συνελικτικών νευρωνικών δικτύων.

Η ανίχνευση όγκου στον εγκέφαλο είναι ένας κύριως στόχος των υπολογιστικών συστημάτων στον τομέα της ιατρικής και πιο συγκεκριμένα στην νευροχειρουργική. Ένας όγκος εγκεφάλου, όπως συμβαίνει με κάθε όγκο στο ανθρώπινο σώμα διακρίνεται σε δύο κύριες κατηγορίες, αυτές είναι : ο καλοήθης (LGG - χαμηλού βαθμού γλοιώματος) και ο κακοήθης (HGG - υψηλού βαθμού γλοιώματος) . Πιο συγκεκριμένα, καλοήθης λέγεται όταν αναπτύσσεται αργά και χωρίς να εισχωρεί στον εγκέφαλο ενώ, ένας όγκος εγκεφάλου λέγεται κακοήθης όταν αναπτύσσεται με γρήγορο ρυθμό και εισχωρεί στον εγκέφαλο που βρίσκεται γύρω του. Η κατάσταση στην οποία βρίσκεται ο όγκος είναι ζωτικής σημασίας για τη θεραπεία. Για την υποστήριξη των νευροχειρούργων κατά τη διάρκεια αυτής της εξέτασης, θα εφαρμοστεί ένα βαθύ συνελικτικό νευρικό δίκτυο για ανίχνευση γλοιώματος με την χρήση ενός υπολογιστικού συστήματος. Σε αυτή την εργασία εφαρμόστηκε η αρχιτεκτονική Faster R-CNN. Για την εκπαίδευση χρησιμοποιήθηκε το dataset του διαγωνισμού BRATS[1]2015 (Brain Tumor Image Segmentation) .Τέλος, δεν έχει τόση σημασία αν είναι καλοήθεις ή κακοήθεις. Ένας όγκος εγκεφάλου, μπορεί να είναι καλοήθης αλλά η εντόπισή του να προκαλεί εξίσου σοβαρά συμπτώματα και να θέτει σε κίνδυνο ακόμα και τη ζωή του πάσχοντος. Συνεπώς, και οι κακοήθεις, αλλά και οι καλοήθεις όγκοι χρειάζονται πάντα αναγνώριση και άμεση αντιμετώπιση.

1.2 Λεπτομέρειες όγκου στον εγκέφαλο (Brain tumor details LGG-HGG)

Η απεικόνιση της μαγνητικής τομογραφίας (MRI) έχει γίνει μια σύγχρονη τεχνική για τη διάγνωση όγκων του εγκεφάλου τις τελευταίες δεκαετίες, λόγω της βελτιωμένης αντίθεσης των μαλακών ιστών. Τα γλοιώματα αποτελούν περίπου το 80% όλων των κακοήθων όγκων του εγκεφάλου που προέρχονται από τα νευρογλοιακά κύτταρα του κεντρικού νευρικού συστήματος. Με βάση την επιθετικότητα των γλοιωμάτων, η Παγκόσμια Οργάνωση Υγείας (WHO) τα κατάταξε σε δύο κατηγορίες. Τα γλοιώματα χαμηλής ποιότητας (LGG), που αποτελούνται από γλοιώματα χαμηλού βαθμού και μεσαίου βαθμού και τα γλοιώματα υψηλού βαθμού (HGG) (Louis, David N., 2016). Τα LGG είναι διεισδυτικά εγκεφαλικά νεοπλάσματα που περιλαμβάνουν ιστολογικές τάξεις, 2 από τις οποίες είναι: τα νεοπλάσματα βαθμού II και III. Παρόλο που οι ασθενείς με LGG είναι σε σχετικά καλύτερη κατάσταση από εκείνους με HGG, οι LGG ασθενείς βρέθηκαν να προχωρούν συνήθως σε δευτερογενή γλοιωβλάστωμα(γλοιώματα υψηλού βαθμού κακοήθειας). Και στις δύο περιπτώσεις είναι αναγκαία η σωστή αντιμετώπιση της θεραπείας, δεδομένου ότι η έγκαιρη και η σωστή ανίχνευση του βαθμού όγκου μπορεί να οδηγήσει σε καλή πρόγνωση και αντιμετώπιση (Van den Bent, M. J., Brandes, 2013).

Η ιστολογική ταξινόμηση, όσον αφορά τη χειρουργική βιοψία, χρησιμοποιείται κυρίως για τη διαχείριση των γλοιωμάτων. Γενικά, το συστατικό υψηλότερης ποιότητας, μεταξύ των παραγόμενων δειγμάτων ιστοπαθολογίας, χρησιμοποιείται για την πρόβλεψη του συνολικού βαθμού όγκου. Τα γλοίωμα που είναι ετερογενή, μερικές φορές λόγο των ιστοπαθολογικών δειγμάτων που συλλέγονται από διαφορετικά μέρη του ίδιου όγκου εμφανίζουν διαφορετικούς βαθμούς. Επειδή οι παθολόγοι δεν παρέχονται με ολόκληρο τον περιγραφόμενο όγκο κατά τη διάρκεια της εξέτασης, είναι πιθανό ότι το συστατικό υψηλότερης ποιότητας μπορεί να λείπει στο δείγμα βιοψίας. Αυτό ονομάζεται σφάλμα δειγματοληψίας βιοψίας (Jackson, R. J., Fuller, 2001) και ενδέχεται να οδηγήσει σε λανθασμένη κλινική αντιμετώπιση της νόσου. Επιπλέον, υπάρχουν αρκετοί παράγοντες κινδύνου στη δοκιμή βιοψίας, συμπεριλαμβανομένης της αιμορραγίας από τον όγκο και τον εγκέφαλο λόγω της βελόνας βιοψίας. προκαλώντας σοβαρή ημικρανία, εγκεφαλικό επεισόδιο, κώμα και ακόμη και θάνατο. Άλλοι σχετικοί κίνδυνοι περιλαμβάνουν τη μόλυνση ή τις επιληπτικές κρίσεις (McGirt, Matthew J., 2005).

1.3 Η απεικόνιση MR (MR imaging)

Η MR απεικόνιση, είναι σε θέση να ανιχνεύει ολόκληρο τον όγκο και μπορεί να επιδείξει μεγάλη συσχέτιση με ιστολογική ποιότητα. Επίσης, δεν είναι ευαίσθητο στο σφάλμα δειγματοληψίας και στη μεταβλητότητα μεταξύ των παραμέτρων και μεταξύ των παρατηρητών. Σε αυτό το πλαίσιο, η MRI πολλαπλών ακολουθιών παίζει σημαντικό ρόλο στην ανίχνευση, τη διάγνωση και τη διαχείριση των καρκίνων του εγκεφάλου. Πρόσφατες αναφορές από τη βιβλιογραφία δείχνουν ότι

η αυτοματοποιημένη ανίχνευση και διάγνωση της νόσου, με βάση την ανάλυση ιατρικών εικόνων, θα μπορούσε να αποτελέσει καλή εναλλακτική λύση. Η αποκωδικοποίηση του φαινοτύπου του όγκου χρησιμοποιώντας μη επεμβατική απεικόνιση είναι ένα πρόσφατο πεδίο έρευνας, γνωστό ως Radiomics (Mitra, S. & Uma Shankar, 2015), και περιλαμβάνει την εξαγωγή ενός μεγάλου αριθμού χαρακτηριστικών απεικόνισης που μπορεί να μην είναι εμφανή στο ανθρώπινο μάτι. Ένα αναπόσπαστο μέρος της διαδικασίας περιλαμβάνει τη χειροκίνητη ή αυτοματοποιημένη οριοθέτηση της 2D περιοχής ενδιαφέροντος (ROI) ή του 3D όγκου ενδιαφέροντος (VOI), για να εστιάσει την προσοχή στην κακοήθη ανάπτυξη. Στη συνέχεια, ακολουθείται από την εξαγωγή κατάλληλων συνόλων χαρακτηριστικών απεικόνισης, τα οποία αναλύονται μέσω μηχανικής μάθησης για τη λήψη αποφάσεων.

1.4 Δομή της διπλωματικής

Αρχικά, στο 1ο κεφάλαιο αναφέρουμε περιληπτικά τις κύριες διαφορές ανάμεσα στους (LGG-χαμηλού βαθμού γλοιώματος) και (HGG-υψηλού βαθμού γλοιώματος) όγκους, καθώς επίσης και μερικές πληροφορίες σχετικά με την απεικόνιση μαγνητικών τομογραφιών. Στο 2ο κεφάλαιο θα αποσαφηνιστούν κάποιες βασικές αρχές. Αυτές οι αρχές αφορούν την ταξινόμηση, τον ταξινομητής αντικειμένου, την ανίχνευση αντικειμένου, την επεξεργασία των εικόνων, τη συλλογή δεδομένων και τη δημιουργία ενός μοντέλου το οποίο συνδυάζει τα χρήσιμα χαρακτηριστικά που εξάγονται με κάποιους αλγορίθμους από κάποιο dataset. Στο 3ο κεφάλαιο θα αναφερθούμε στους παράγοντες που δυσκολεύουν την αναγνώριση αντικειμένου αλλά και στις προσεγγίσεις αναγνώρισης του αντικειμένου. Κατόπιν, στο 4ο κεφάλαιο γίνεται αναφορά στους ανιχνευτές υψηλής τεχνολογίας CNN (Faster R-CNN, YOLO, SSD). Στο κεφάλαιο 5 γίνεται αναφορά στα εργαλεία που χρησιμοποιήθηκαν (python, OpenCV, το Tensorflow, το CUDA και την βιβλιοθήκη Nvidia Cudnn).Στο κεφάλαιο 6 γίνεται η υλοποίηση του νευρωνικού δικτύου, με σκοπό την αναγνώριση όγκου στον εγκέφαλο. Στην συνέχεια θα αναφερθούν συνοπτικά τα βήματα της υλοποίησης. Τέλος, στο κεφάλαιο 7 είναι τα συμπεράσματα που βγάλαμε μετά από την υλοποίηση και εκπαίδευση του νευρωνικού μας δικτύου, ενώ στο κεφάλαιο 8ο αναφέρεται η μελλοντική εργασία πάνω στο μοντέλο που έχουμε υλοποιήσει .

1. http://braintumorsegmentation.org

2 *Βασικές επιστημονικές αρχές*

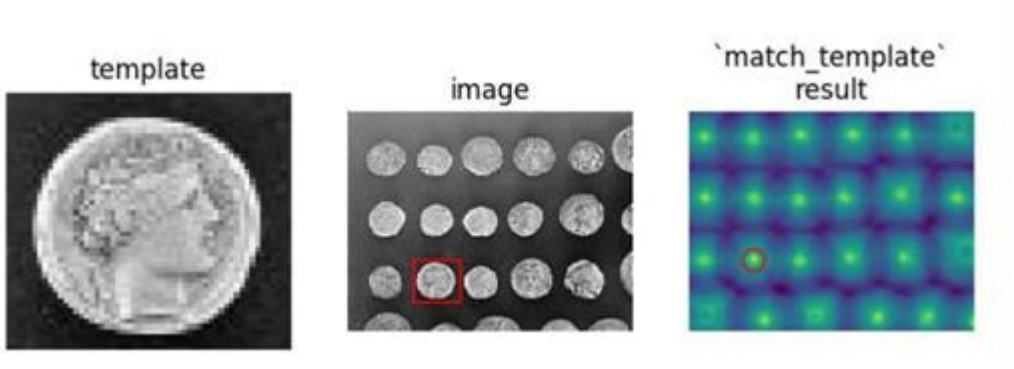
Η αναγνώριση αντικειμένων είναι μια εντυπωσιακή ικανότητα των ανθρώπων. Με μια ματιά ενός αντικειμένου, οι άνθρωποι είναι σε θέση να αναγνωρίσουν την ταυτότητα και την κατηγορία του, παρά την παραμόρφωση τους που οφείλεται στην αλλαγή της θέσης, του φωτισμού και της υφής. Γι' αυτό το λόγο δημιουργήθηκε η ανάγκη να αναπτυχθούν υπολογιστικά συστήματα με αυτήν την ικανότητα. Στη συνέχεια αναφέρονται βασικές αρχές, αλγόριθμοι και δυσκολίες για την υλοποίηση της ικανότητας αυτής από τους υπολογιστές.

2.1 Ταξινόμηση (Classification)

Πρόκειται για μια διαδικασία που σχετίζεται με το categorization (κατηγοριοποίηση) και αφορά την αναγνώριση και κατηγοριοποίηση αντικειμένων με βάση κάποια συγκεκριμένα χαρακτηριστικά. Βασιζόμενοι στα χαρακτηριστικά (features) που έχουμε εξάγει από το σύνολο των αντικειμένων (dataset), οι προσεγγίσεις με classification μπορούν να έχουν τα καλύτερα αποτελέσματα. Τρείς είναι οι πιο συνηθισμένες μέθοδοι (περαιτέρω ανάλυση τους στο κεφάλαιο 3).

Template Matching

Αυτή η μέθοδος χρησιμοποιείται κυρίως για αναζήτηση μέγιστης αντιστοίχισης (maximum matching) (Huang, Jiang & Klette, 2009). Όλα τα απαιτούμενα χαρακτηρίστηκα του object που πρέπει να κατηγοριοποιηθούν αποθηκεύονται σε μια βάση δεδομένων, συμπεριλαμβανομένων σημαντικών διαφορών στην απόδοση (performance), το σημείο θέασης (viewpoint) και τον φωτισμό (illumination). Κάθε αντικείμενο συγκρίνεται με κάθε πρότυπο (template) ιδίων χαρακτηριστικών (features). Η αντιστοίχιση των προτύπων είναι γρήγορη και εύκολη να τροποποιηθεί για νέες κλάσεις.

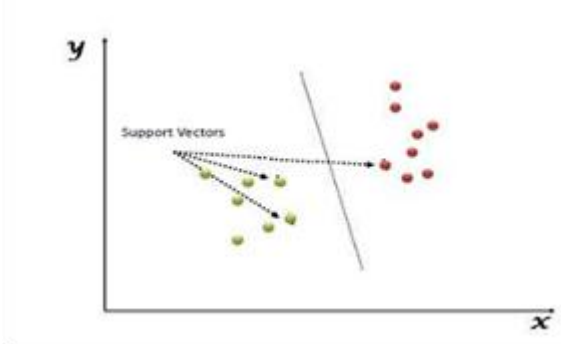


Εικόνα 1:Template Matching

Support Vector Machine

Τα "Support Vector Machine" (SVM) είναι ένας supervised machine learning αλγόριθμος ο οποίος μπορεί να χρησιμοποιηθεί τόσο για το classification όσο και για regression challenges. Ωστόσο χρησιμοποιείται κυρίως σε προβλήματα ταξινόμησης. Σε αυτόν τον αλγόριθμο σχεδιάζουμε κάθε στοιχείο δεδομένων ως σημείο σε n-διάστατο χώρο με την αξία του κάθε χαρακτηριστικού να είναι η τιμή μιας συγκεκριμένης συντεταγμένης. Στην συνέχεια ταξινομούμε βρίσκοντας το hyper-plane που διαφοροποιεί τις 2 κατηγορίες.

Το "Support Vector" είναι στην ουσία οι συντεταγμένες που παρατηρήθηκαν, καθώς το Support Vector Machine είναι το όριο που διαχωρίζει αυτές τις 2 κατηγορίες (Li &Tang, 2004).

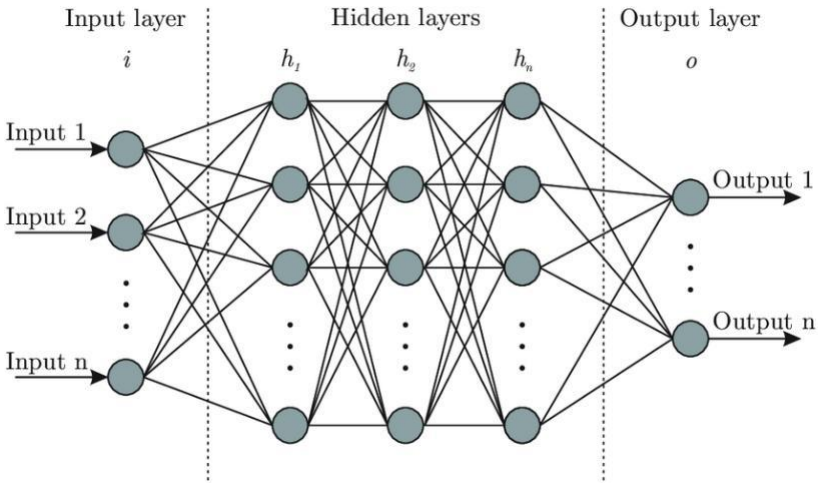


Εικόνα 2: SVC

Artificial Neural Network Algorithms

Ένα τεχνητό νευρωνικό δίκτυο (ANN) είναι ένα παράδειγμα επεξεργασίας πληροφοριών που εμπνέεται από τον τρόπο με τον οποίο τα βιολογικά νευρικά συστήματα (όπως ο εγκέφαλος) επεξεργάζονται πληροφορίες. Το βασικό τους χαρακτηριστικό είναι η νέα

δομή επεξεργασίας πληροφοριών. Αποτελούνται από έναν μεγάλο αριθμό αλληλοσυνδεδεμένων στοιχείων επεξεργασίας (νευρώνες) που λειτουργούν από κοινού για την επίλυση συγκεκριμένων προβλημάτων. Τα ANN όπως και ο άνθρωπος μπορούν να μάθουν από ένα παράδειγμα και από το λάθος τους (μέχρι να ελαχιστοποιηθεί η συνάρτηση λάθους). Ένα ANN μπορεί να ρυθμιστεί για μια συγκεκριμένη εφαρμογή πχ. Object recognition μέσω μιας διαδικασίας εκμάθησης. Όπου στην ουσία εξάγοντας τα χρήσιμα δεδομένα από κάποια σετ δεδομένων (dataset), εκπαιδεύονται και έτσι στην συγκεκριμένη περίπτωση μπορούν να εντοπίσουν αντικείμενα και να τα κατηγοριοποιήσουν (Torres, Hervás & García, 2009).



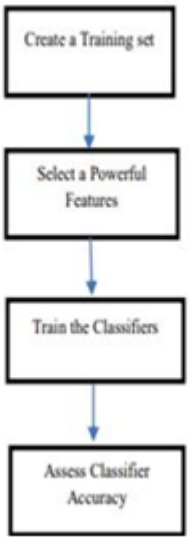
Εικόνα 3: Artificial Neural Network Architecture

2.2 Ταξινομητής αντικειμένου (Object Classifier)

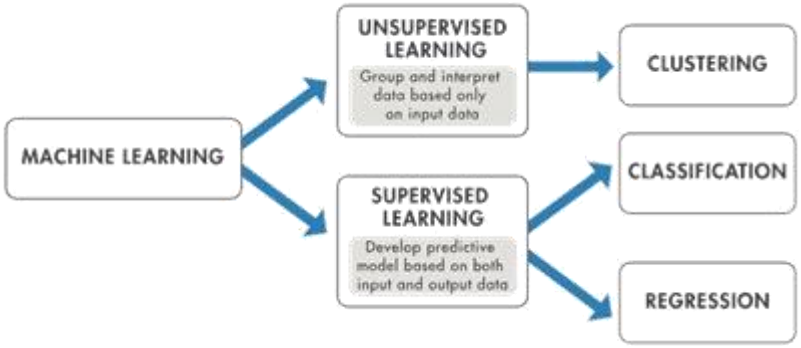
Ένας ταξινομητής (classifier) είναι ένας αλγόριθμος που παίρνει ένα σύνολο χαρακτηριστικών (features) που χαρακτηρίζουν αντικείμενα και τα χρησιμοποιεί για τον καθορισμό της κλάσης ενός αντικειμένου. Για κάθε αντικείμενο ένας αλγόριθμος feature extraction παίρνει κάποια χαρακτηριστικά όπως (size, compactness, boundary box etc), έπειτα, ο classifier χρησιμοποιεί αυτές τι ιδιότητες-χαρακτηριστικά για να αποφασίσει σε ποια κλάση ανήκει το ζητούμενο αντικείμενο.

Υπάρχουν 2 είδη classifier οι supervised και οι unsupervised (Zhang, Tianzhu, Si Liu, Changsheng Xu, and Hanqing Lu, 2013). Το supervised learning, σημαίνει ότι κάποιος ειδικός έχει ορίσει τις κλάσεις που αντιπροσωπεύουν τα αντικείμενα. Όπως επίσης έχει ορίσει και ένα dataset από αντικείμενα με τις γνωστές κλάσεις τους. Αυτά τα dataset ονομάζονται και training sets. Τα training set χρησιμοποιούνται από τους classifier για να μάθουν να ταξινομούν αντικείμενα. Στην unsupervised learning ο μηχανισμός λειτουργεί απευθείας από τα δεδομένα και δεν υπάρχουν σύνολα εκπαίδευσης ούτε προκαθορισμένες κατηγορίες δηλαδή η μη παρακολουθούμενη μάθηση είναι όπου υπάρχουν μόνο τα δεδομένα εισόδου και δεν υπάρχει αντίστοιχη μεταβλητή εξόδου. Υπάρχουν τέσσερα βήματα για την ανάπτυξη classifiers:

- Δημιουργία του συνόλου εκπαίδευσης.
- Επιλογή χαρακτηριστικών.
- Εκπαίδευση του classifier.
- Αξιολόγηση της ακρίβειας του classifier.



Εικόνα 4: Object Classifier



Εικόνα 5: Supervised and Unsupervised learning

2.3 Ανίχνευση αντικειμένου (Object Detection)

Η ανίχνευση αντικειμένων είναι μια τεχνολογία των υπολογιστών που σχετίζεται με την όραση και την επεξεργασία των εικόνων που αφορά την ανίχνευση αντικειμένων μιας συγκεκριμένης κλάσης (όπως άνθρωποι, κτίρια ή αυτοκίνητα) σε εικόνες και βίντεο. Η ανίχνευση αντικειμένων έχει εφαρμογές σε πολλούς τομείς της υπολογιστικής όρασης, συμπεριλαμβανομένης της ανάκτησης εικόνων και της επιτήρησης βίντεο. Η ανίχνευση είναι η διαδικασία εύρεσης αντικειμένου σε ένα μόνο πλαίσιο, δηλαδή αποτελεί ένα instance του object tracking που αναλύεται στην συνέχεια.

Τα οπτικά χαρακτηριστικά χαμηλής στάθμης μιας εικόνας, όπως ένας χάρτης ορατότητας, μπορούν να χρησιμοποιηθούν ως οδηγός για τον εντοπισμό υποψήφιων αντικειμένων (Walther, Itti, Riesenhuber, Poggio & Koch, 2002). Η τοποθεσία και το μέγεθος καθορίζονται συνήθως χρησιμοποιώντας ένα πλαίσιο οριοθέτησης, το οποίο αποθηκεύεται με τη μορφή συντεταγμένων. Η εικόνα που περιέχεται στο πλαίσιο οριοθέτησης ταξινομείται στη συνέχεια από έναν αλγόριθμο που έχει εκπαιδευτεί με τη χρήση μηχανικής μάθησης (Girshick, Donahue, Darrell, Malik & Rich, 2014). Τα όρια του αντικειμένου μπορούν να βελτιωθούν περαιτέρω επαναληπτικά.

2.4 Παρακολούθηση αντικειμένων (Object Tracking)

Κατά τη διαδικασία παρακολούθησης ενός αντικειμένου ακολουθούνται βασικά βήματα. Αρχικά, η είσοδος αποτελείται από μια πεπερασμένη ακολουθία πινάκων (Mo, ..., Mn) για κάποιο n ∈ N (ένας μοναδικός πίνακας αντιπροσωπεύει μία εικόνα στην ροή βίντεο που λαμβάνεται από την κάμερα και οι καταχωρήσεις του αντιπροσωπεύουν μεμονωμένα pixel) και ένα αρχικό υποσύνολο επιλεγμένων καταχωρήσεων από τον πίνακα Mo, ο οποίος αντιπροσωπεύει το αντικείμενο ενδιαφέροντος. Στόχος αυτής της διαδικασίας είναι να ακολουθήσουμε αυτό το αντικείμενο μέσω της σειράς εικόνων και για κάθε εικόνα Mi εκπέμπεται ένα υποσύνολο των καταχωρήσεων Mi που αντιστοιχούν στο αντικείμενο ενδιαφέροντος (ίσως με αλλαγμένη εμφάνιση) από την εικόνα Mo. Το μεγαλύτερο εμπόδιο με αυτήν την περιγραφή προβλημάτων είναι το γεγονός ότι το αντικείμενο του ενδιαφέροντος δεν έχει καθοριστεί καλά, με εξαίρεση την αρχική εικόνα (λόγω εξωτερικών συνθηκών) (Russell, Stuart, Norvig, Peter, 2009). Ακόμη και στις πιο ιδανικές συνθήκες, οι ανιχνευτές πρέπει να ξεπεράσουν μια μεγάλη ποικιλία εμποδίων για να αποδώσουν καλά. Πράγματι, ορίζονται αρκετά διαφορετικά "χαρακτηριστικά" των ακολουθιών του βίντεο που μπορούν να παρεμποδίσουν την απόδοση της παρακολούθησης. Μεταξύ αυτών είναι η περιστροφή τόσο μέσα όσο και έξω από το επίπεδο της εικόνας, και η ανομοιομορφία του περιβάλλοντος κ.λπ.. (Russell, Stuart, Norvig, Peter, 2009).

Για να επιτευχθεί η παρακολούθηση θα πρέπει ο ανιχνευτής να δημιουργήσει ένα μοντέλο που θα ενσωματώνει πολλές όψεις του στόχου. Ωστόσο, ένα δυναμικό μοντέλο έχει μεγάλη πιθανότητα λάθους καθώς μπορεί να παρασυρθεί, πράγμα που σημαίνει ότι αρχίζει να ενσωματώνει πληροφορίες φόντου μέχρι να χάσει τελείως τον στόχο. Με αυτή την έννοια, η μεγαλύτερη πρόκληση της διαδικασία παρακολούθησης είναι η δημιουργία ενός αλγορίθμου που θα μπορεί να αλλάξει τη στοχευμένη εμφάνιση, ενώ, παράλληλα, θα εξαλείφει την παρασυρόμενη κίνηση.

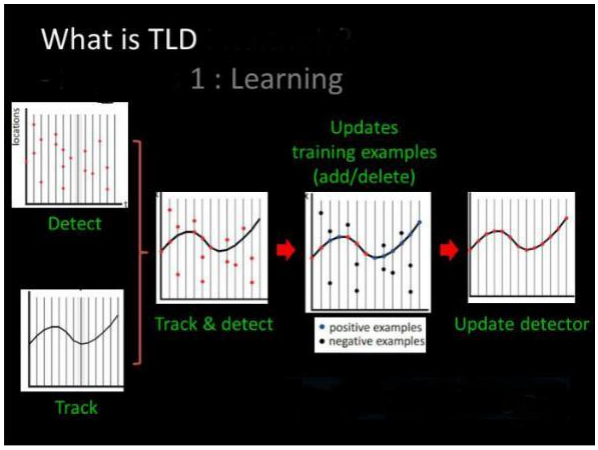
Μια από τις πιο διαδεδομένες αρχιτεκτονικές παρακολούθησης είναι αυτή που βασίζεται στον αλγόριθμο ανίχνευσης εκμάθησης εντοπισμού (TLD-Tracking Learning Detection) (Z. Kalal, K. Mikolajczyk, and J. Matas, 2010). Εν μέρει βασίζεται στις τεχνικές επεξεργασίας εικόνας και στις δυνατότητες δημιουργίας πλαισίων οριοθέτησης, αλλά επίσης κάνει βαριά χρήση της μηχανικής μάθησης. Όπως υποδηλώνει το όνομα, στο TLD η μακροπρόθεσμη ανίχνευση αντικειμένων αποτελείται από τρία στάδια:

1. Παρακολούθηση (tracking): Το πρόβλημα της εύρεσης αντικειμένου σε ένα νέο πλαίσιο εικόνων δεδομένων της θέσης του σε ένα προηγούμενο πλαίσιο.
2. Ανίχνευση (detection): Το πρόβλημα της εύρεσης ενός αντικειμένου σε μια εικόνα, δεδομένου ενός μοντέλου αυτού του αντικειμένου.
3. Μάθηση (learning): Το πρόβλημα της ενημέρωσης του αλγορίθμου ανίχνευσης για την ενσωμάτωση νέων προβολών του προς παρακολούθηση αντικειμένου.

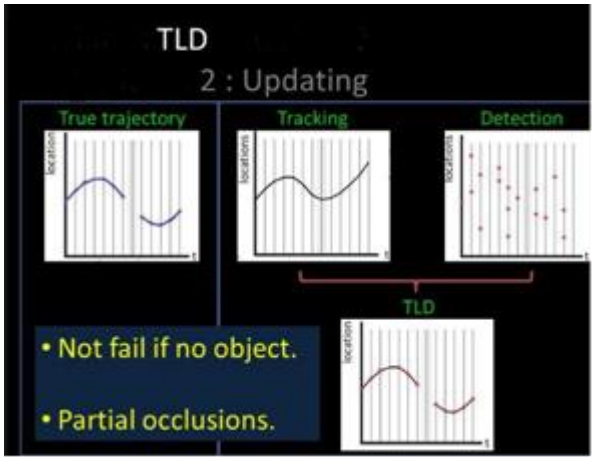
Αρχικά, η διαφορά μεταξύ παρακολούθησης και ανίχνευσης μπορεί να φαίνεται τετριμμένη. Ωστόσο, όπως επισημαίνει ο Kalal (2010), οι δύο εργασίες είναι διακριτές, επειδή η ανίχνευση περιλαμβάνει εντοπισμό ενός αντικειμένου εντός μιας εικόνας βασισμένη σε ένα μοντέλο αντικειμένου, ενώ η παρακολούθηση χρησιμοποιεί πληροφορίες σχετικά με τη θέση στόχου σε προηγούμενα πλαίσια για να καθορίσει τη θέση στόχου στο τρέχον πλαίσιο. Κατά συνέπεια, η παρακολούθηση και η ανίχνευση είναι επιρρεπείς σε διαφορετικά είδη σφαλμάτων: οι ανιχνευτές τείνουν να παρασύρονται από τους στόχους τους και σχεδόν βέβαια αποτυγχάνουν όταν ο στόχος τους αφήνει το πλαίσιο και επιστρέφει. Οι ανιχνευτές, από την άλλη πλευρά, θα εντοπίζουν πάντα σωστά το αντικείμενο, υποθέτοντας ότι ταιριάζει με το μοντέλο αντικειμένου με κάποιο βαθμό εμπιστοσύνης. Εντούτοις, επειδή τα αντικειμενικά μοντέλα είναι γενικά στατικά, οι ανιχνευτές αποτυγχάνουν όταν η εμφάνιση του αντικειμένου αλλάζει μέσω περιστροφής, παραμόρφωσης κλπ. Η βασική ιδέα του TLD είναι να εκπαιδευτεί με επαναληπτικό τρόπο ο αλγόριθμος ανίχνευσης χρησιμοποιώντας ένα σύνολο εκπαίδευσης που ενημερώνεται συνεχώς με προηγουμένως εσφαλμένες (false positive or false negative) εικόνες. Ο Kalal χρησιμοποιεί ένα ζευγάρι συστημάτων εμπειρογνωμόνων, το ένα για την εύρεση false positives και το άλλο για την εύρεση false negative, για να προσθέσει νέες ετικέτες εικόνας στο σύνολο εκπαίδευσης σε κάθε βήμα του χρόνου. Ο ανιχνευτής υλοποιείται ως ένα σύνολο Bayesian ταξινομητών και ένας ταξινομητής πλησιέστερου γείτονα που επανεκπαιδεύεται στο σύνολο εκπαίδευσης σε τακτά χρονικά διαστήματα.

Για το στοιχείο παρακολούθησης, ο Kalal, (2010) χρησιμοποιεί τον ανιχνευτή μέσης ροής που ανέπτυξε (Median-Flow tracker) (Z. Kalal, K. Mikolajczyk, and J. Matas, 2010). Ο ανιχνευτής μέσων ροών χρησιμοποιεί την έννοια της οπτικής ροής (B. K. Horn and B. G. Schunck, 1981) για να προσδιορίσει την κίνηση πλαισίου-πλαισίου των σημείων-κλειδιών στο αντικείμενο. Ο ανιχνευτής χρησιμοποιεί οπτική ροή για την πρόβλεψη της νέας θέσης των σημείων-κλειδιών σε κάθε πλαίσιο και στη συνέχεια εξαλείφει μερικά από αυτά με βάση ένα μετρητή εμπιστοσύνης. Δεδομένου ότι ο ανιχνευτής είναι επιρρεπής στη μετατόπιση και στο

σφάλμα, ο ανιχνευτής χρησιμοποιείται για την εκ νέου ενεργοποίηση του ανιχνευτή εάν κινείται πολύ μακριά από τον στόχο.



Εικόνα 6: TLD Learning



Εικόνα 7: TLD Updating

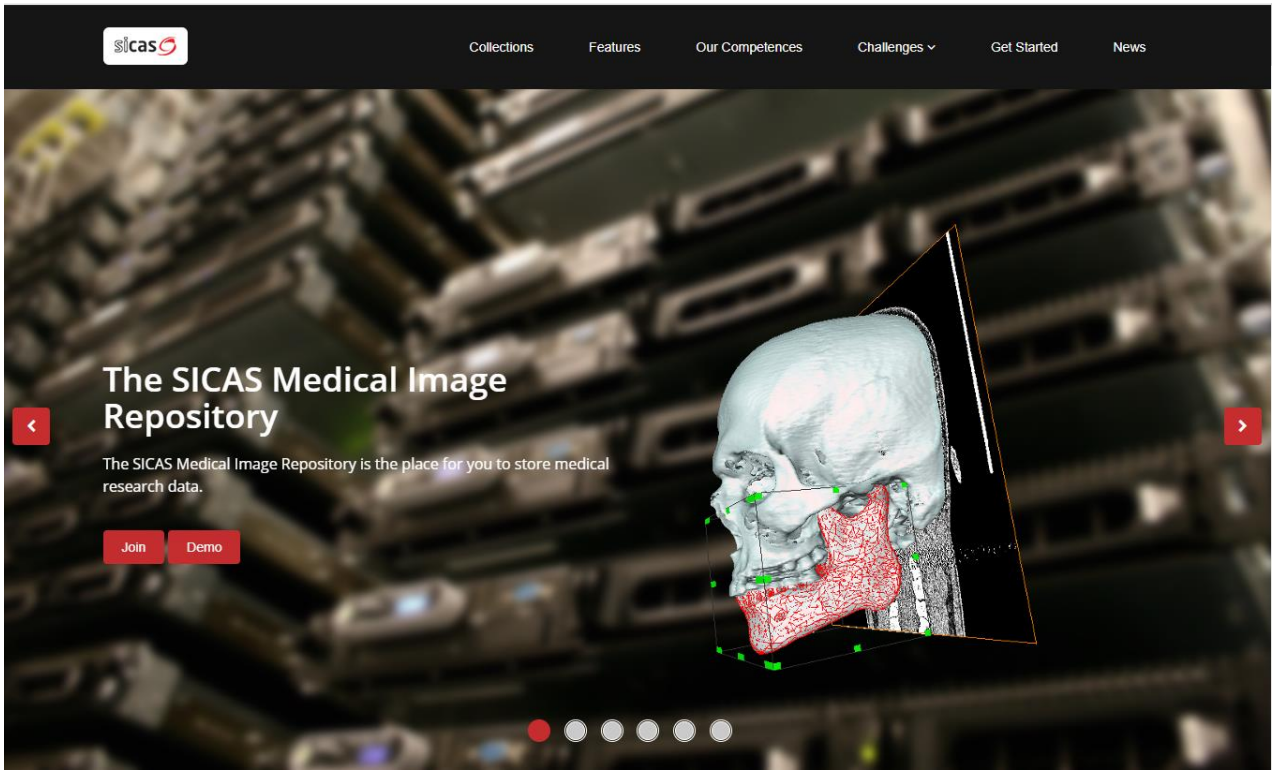
Ένα πιο πρόσφατο παράδειγμα ενός μοντέλου υψηλής απόδοσης είναι το μοντέλο, των G. Nebehay και R. Pflugfelder (2015), Clustering of Static-Adaptive Correspondences for Deformable Object Tracking (Ομαδοποίηση Στατικών Προσαρμοζόμενων Αντιστοιχιών για Παρακολούθηση Παραμορφώσιμων Αντικειμένων). Η κεντρική ιδέα του μοντέλου είναι να χρησιμοποιήσει το πρώτο πλαίσιο για να βρει ένα σύνολο αντικειμένων "keypoints" χρησιμοποιώντας τον αλγόριθμο BRISK (S. Leutenegger, M. Chli, and R. Y. Siegwart, 2011), ο οποίος αποτελείται από μικρά, αμετάβλητα χαρακτηριστικά του αντικειμένου τα οποία στη συνέχεια παρακολουθούν την κίνηση αυτών των σημείων. Για να γίνει αυτό, ο αλγόριθμος χρησιμοποιεί δύο μοντέλα του αντικειμένου. Ένα από τα μοντέλα είναι στατικό, ενώ το άλλο ενημερώνεται δυναμικά κατά την παρακολούθηση με νέες πληροφορίες σχετικά με το αντικείμενο. Το στατικό μοντέλο είναι απλά το σύνολο των σημείων-κλειδιών που ανακαλύφθηκαν στο πρώτο πλαίσιο. Αυτή είναι μια αξιόπιστη αναπαράσταση του αντικειμένου επειδή είναι επιλεγμένη από τον χρήστη και δεν θα παρασυρθεί, αφού δεν ενημερώνεται ποτέ. Το δυναμικό μοντέλο είναι το σύνολο των βασικών σημείων στόχου στο προηγούμενο πλαίσιο. Αυτό το μοντέλο είναι απαραίτητο για την αντιμετώπιση της κλιμάκωσης (scaling), της περιστροφής (rotation), της ορατότητας (occlusion) και της παραμόρφωσης του αντικειμένου (deformation).

Σε κάθε βήμα του χρόνου, χρησιμοποιείται η ακόλουθη διαδικασία για την εύρεση της νέας θέσης στόχου: η συνολική εικόνα αναζητείται για σημεία-κλειδιά που αντιστοιχούν στα σημεία-κλειδιά του στατικού μοντέλου. Ένα σημείο στο τρέχον πλαίσιο θεωρείται ότι συμφωνεί με ένα σημείο στο στατικό μοντέλο εάν η απόσταση μεταξύ τους είναι κάτω από ένα συγκεκριμένο όριο και είναι πολύ πιο κοντά μεταξύ τους από οποιοδήποτε άλλο από τα υποψήφια σημεία-κλειδιά. Οι αντιστοιχίες στο προσαρμοστικό μοντέλο εντοπίζονται χρησιμοποιώντας την οπτική ροή. Εάν τα δύο στατικά και τα προσαρμοστικά μοντέλα ταιριάζουν με το ίδιο σημείο κλειδιού, η στατική αντιστοίχιση υπερισχύει, καθώς το στατικό μοντέλο δεν υπόκειται σε μετατόπιση. Τα υποψήφια σημεία-κλειδιά ομαδοποιούνται έπειτα χρησιμοποιώντας έναν αλγόριθμο συστάδων (clustering) και τέλος υπολογίζεται η κλίμακα (scaling) του μοντέλου, που είναι, για όλα τα ζεύγη σημείων αντιστοίχισης, η μέση αναλογία της απόστασης μεταξύ των σημείων στο αρχικό πλαίσιο και στο τρέχον πλαίσιο. Διαισθητικά, αυτό θα πρέπει να είναι ισχυρό στις τοπικές παραμορφώσεις στο αντικείμενο (ακόμη και σε μικρές), επειδή ο διάμεσος δεν επηρεάζεται από τις υπερβολικές τιμές. Ωστόσο, θα πρέπει επίσης να συλλάβει τις παραμορφώσεις που επηρεάζουν το συνολικό μέγεθος του αντικειμένου, επειδή το συνολικό μέγεθος θα αλλάξει για όλα τα σημεία.

2.5 Συλλογή δεδομένων (The dataset)

Ένα dataset αποτελεί μια συλλογή από δεδομένα. Στις μέρες μας υπάρχουν αρκετά έτοιμα data-set που το καθένα εξυπηρετεί το δικό του σκοπό (π.χ. ImageNet, Ms Coco, Mnist, Smir κ.τ.λ.). χαρακτηριστικό παράδειγμα αποτελεί το Smir[2] (Sicas Medical Image Repository), είναι μια μεγάλη βάση ιατρικών δεδομένων σχεδιασμένη για χρήση στην ιατρική έρευνα. Από τα dataset εξάγουν τα

features που χρειάζονται για την διαδικασία του object recognition. Όσο πιο ποιοτικό είναι ένα dataset τόσο πιο ποιοτικά είναι τα features που θα εξαχθούν από τον αλγόριθμο και κατά συνέπεια τόσο πιο ακριβές θα είναι το μοντέλο που θα δημιουργηθεί. Με τον όρο ποιοτικό εννοούμε δεδομένα όπως την ανάλυση της εικόνας, το μέγεθος αυτής, τα δεδομένα που εμπεριέχει κ.τ.λ. Τέλος τα dataset παίζουν, αν όχι τον σημαντικότερο αλλά από τους σημαντικότερους ρόλους σε ένα σύστημα object recognition που έχει υλοποιηθεί με την τεχνική του Deep learning διότι αρχικά όσο περισσότερα δεδομένα εμπεριέχονται στην εξίσωση τόσο το καλύτερο αποτέλεσμα θα πάρουμε. Αυτή είναι και η λογική άλλωστε του Machine learning με τον τρόπο που χρησιμοποιείται στο Υπολογιστική όραση, η εξαγωγή συμπερασμάτων από πολλά δεδομένα. Όσο μεγαλύτερο λοιπόν και ακριβέστερο ένα dataset τόσο το καλύτερο ειδικά για προσεγγίσεις μέσω του Machine Learning.



Εικόνα 8: Smir

2. https://www.smir.ch

2.6 Το μοντέλο (Model)

Ένα μοντέλο συνδυάζει τα χρήσιμα χαρακτηριστικά που εξάγονται με κάποιος αλγορίθμους από κάποιο dataset, για την χρήση αυτών στην εξαγωγή αποφάσεων, αναγνώρισης αντικειμένων κτλ. Ένα μοντέλο από την οπτική γωνία του Machine Learning είναι στην ουσία ένα instance ενός Νευρωνικού δικτύου, που έχει εκπαιδευτεί αντλώντας δεδομένα από κάποιο dataset, για την εξαγωγή κάποιον συμπερασμάτων, την αναγνώριση – κατηγοριοποίηση των δεδομένων.

2.7 Παράγοντες που δυσκολεύουν την αναγνώριση αντικειμένου

- **Φωτισμός (illumination):** Οι συνθήκες του φωτισμού ποικίλουν μέσα στην πορεία της ημέρας. Επιπλέον, λόγω των καιρικών συνθηκών δημιουργείται διαφορετικός φωτισμός στο χώρο, για παράδειγμα εικόνες τραβηγμένες σε εξωτερικό χώρο διαφέρουν από τις εικόνες που τραβήχτηκαν σε εσωτερικό χώρο. Γι' αυτό το λόγο, ένα σύστημα πρέπει να είναι σε θέση να αναγνωρίζει τα αντικείμενα ανεξαρτήτως του φυσικού φωτισμού.

- **Θέση (Positioning):** Η θέση του αντικειμένου στην εικόνα ή διαφορετικά το λεγόμενο σημείο θέασης (point of view). Το σύστημα οφείλει να αναγνωρίζει το αντικείμενο του ενδιαφέροντος ανεξαρτήτως του σημείου θέασης ή της θέσης του αντικειμένου στην εικόνα.

- **Περιστροφή (Rotation):** Μια από τις δυνατότητες που έχουν οι εικόνες είναι η περιστροφή σε σχέση με την αρχική της μορφή ή και το αντικείμενο. Έτσι το σύστημα οφείλει να ανιχνεύει τα αντικείμενα ανεξαρτήτως την φορά ή την διεύθυνση στην οποία βρίσκονται.

- **Καθρέφτισμα (Mirroring):** Το είδωλο από ένα καθρεπτισμένο αντικείμενο πρέπει να μπορεί να ανιχνεύεται από το σύστημα.

- **Ορατότητα (Occlusion):** Στην περίπτωση που το αντικείμενο μιας εικόνας δεν είναι ευδιάκριτο, το σύστημα θα πρέπει να μπορεί να το αναγνωρίσει

- **Κλίμακα (Scale):** Η ορθότητα του συστήματος δεν πρέπει να επηρεάζεται από την αλλαγή στο μέγεθος του αντικειμένου

3 *Προσεγγίσεις στην αναγνώριση αντικειμένου*

Υπάρχουν πολλά προβλήματα που έχουν οι αλγόριθμοι αναγνώρισης και παρακολούθησης αντικειμένων να αντιμετωπίσουν. Ένα από αυτά είναι το πρόβλημα των παραλλαγών εμφάνισης του στόχου όπως παραμόρφωση σχήματος, αλλαγές κλίμακας, αλλαγή φωτισμού ή κίνηση κάμερας. Επίσης, τα αντικείμενα μερικές φορές ξεφεύγουν συχνά εντελώς έξω από τα όρια της εικόνας. Επομένως, είναι απαραίτητο ο αλγόριθμος παρακολούθησης να είναι σε θέση να ανιχνεύσει τον στόχο ανεξάρτητα από οποιαδήποτε προηγούμενη ακολουθία επεξεργασμένων εικόνων μέχρι τώρα. Ένα άλλο πρόβλημα είναι και ο θόρυβος που προκύπτει από τη λειτουργία της κάμερας. Στη συγκεκριμένη εργασία η μέθοδος που θα ακολουθηθεί αφορά την επεξεργασία εικόνας σε πραγματικό χρόνο. Παρακάτω θα γίνει ανάλυση των διάφορων προσεγγίσεων για την αναγνώριση αντικειμένου.

3.1 Ταίριασμα προτύπου

Η αντιστοίχιση προτύπου είναι μια πολύ απλή διαδικασία. Αυτή η μέθοδος ταιριάζει αποθηκευμένες εικόνες προτύπου με μια δεδομένη εικόνα για τον προσδιορισμό αντικειμένων στην εικόνα που εισάγουμε. Αυτό μπορεί να γίνει τόσο σε χρωματικές όσο και σε εικόνες σε κλίμακα του γκρι (Khushboo Khurana, 2013). Αυτή η τεχνική μπορεί είτε να βασίζεται σε αντιστοίχιση pixel με pixel είτε σε αντιστοίχιση χαρακτηριστικών. Στην προσέγγιση αυτή, τα χαρακτηριστικά των εικόνων προτύπων είναι σε σύγκριση με κομμάτια της δεδομένης εικόνας εισόδου (Khushboo Khurana, 2013)

3.2 Μέθοδος βασισμένη στο χρώμα

Μια άλλη απλή και αποτελεσματική τεχνική ανίχνευσης αντικειμένων είναι η αντιστοίχιση εικόνων βάσει των χρωμάτων τους. Ωστόσο, όταν ο φωτισμός ή οι συνθήκες δεν είναι οι ίδιες όπως στις εικόνες προτύπων, η ακριβή αναγνώριση αντικειμένων υποβαθμίζεται σημαντικά. Υπάρχουν επίσης πολλοί χρωματικοί χώροι που μπορούν να χρησιμοποιηθούν. Οι συνήθεις γνωστοί χώροι χρώματος περιλαμβάνουν: RGB (κόκκινο-πράσινο-μπλε), CMY (κυανό-ματζέντα-κίτρινο). Μπορούμε επίσης να χρησιμοποιήσουμε ένταση I, απόχρωση H και κορεσμός S. Στον πίνακα,

μπορούμε να δούμε τις διαφορές στην διακύμανση αυτών των έγχρωμων μοντέλων σε διαφορετικές συνθήκες της εικόνας (T. Gevers and A. W. Smeulders, 1999).

	viewing direction	surface orientation	highlights	illumination direction	illumination intensity
I	-	-	-	-	-
RGB	-	-	-	-	-
rgb	+	+	-	+	+
S	+	+	-	+	+
H	+	+	+	+	+

Εικόνα 9: Πίνακας Διακύμανσης Έγχρωμων Μοντέλων

3.3 Μέθοδος βασισμένη στα χαρακτηριστικά

Οι μέθοδοι βασισμένες στα χαρακτηριστικά συγκρίνουν τα χαρακτηριστικά που εξάγονται από μια εικόνα εισόδου με τα χαρακτηριστικά που έχουν εξαχθεί από τα πρότυπα αντικείμενα (J. W. Howarth, H. H. C. Bakker, and R. C. Flemmer, 2009). Δημοφιλή μηχανήματα εξαγωγής περιλαμβάνουν, για παράδειγμα τον Speeded Up Robust Features extractor (SURF)(H. Bay, T. Tuytelaars, and L. Van Gool, 2006), που έχει χρησιμοποιηθεί για ανίχνευση προσώπου, ή για νευρωνικό δίκτυο εξαγώγής χαρακτηριστικών όπως το Darknet-53, που χρησιμοποιούνται από μερικούς σύγχρονους (state-of-the-art) ανιχνευτές νευρωνικών δικτύων (J. Redmon and A. Farhadi, 2018).

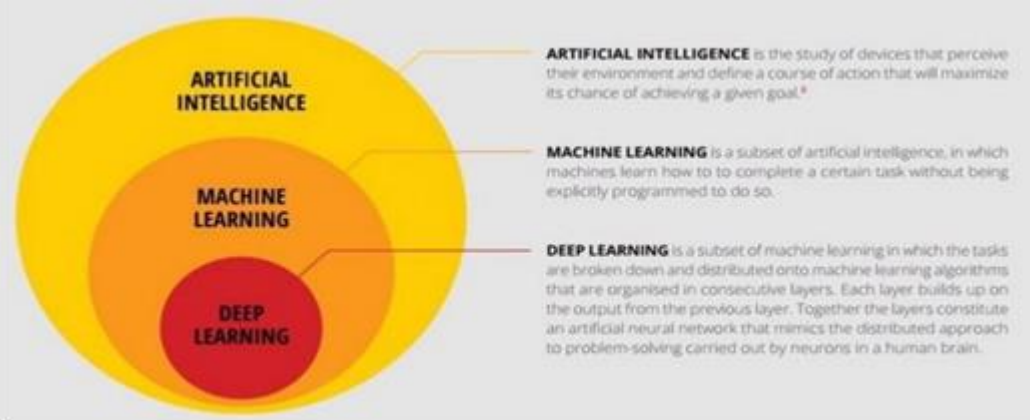
3.4 Διαδοχικοί ταξινομητές (Cascading classifiers)

Αυτό είναι ίσως το καλύτερο, όμως το πλέον υπολογιστικά απαιτητικό αντικείμενο ανίχνευσης-αλγόριθμος που δοκιμάσθηκε. Οι cascade ταξινομητές, συγκολλούν αρκετούς ταξινομητές και μαζί σχηματίζουν μια πολύ καλή συνδυασμένη ταξινόμηση. Οι ταξινομητές αυτοί κατασκευάζονται από βασικούς ταξινομητές αποφάσεων (Russell, Stuart, Norvig, 2009), οι οποίοι τροφοδοτούνται με διάφορα χαρακτηριστικά (π.χ. άκρες, γραμμές κ.λπ..). Είναι δυνατόν να δημιουργηθεί ένας cascade ταξινομητής σε οποιοδήποτε αντικείμενο. Ο ταξινομητής είναι προ-εκπαιδευμένος με μερικές εκατοντάδες δείγματα ενός συγκεκριμένου αντικειμένου που έχουν κλιμακωθεί σε ένα σταθερό μέγεθος και ένα ακόμη μεγαλύτερο σύνολο αρνητικών δείγματων-αυθαίρετες εικόνες που δεν περιέχουν το συγκεκριμένο αντικείμενο. Ένα βασικό ελάττωμα είναι το γεγονός ότι είναι δύσκολο να εκπαιδεύσουμε ένα νέο ανιχνευτή αντικειμένων και μπορεί να

χρειαστούν αρκετές ώρες ή και ακόμη ημέρες για την ολοκλήρωση αυτής της διαδικασίας (περιλαμβάνονται στη βιβλιοθήκη του OpenCV).

3.5 Αναγνώριση αντικειμένου χρησιμοποιώντας μοντέλα βαθιάς μάθησης

Μια πολύ σημαντική εξέλιξη των τελευταίων ετών αφορά την αναγνώριση των αντικειμένων (Object Recognition) ως ένα από τα πιο συναρπαστικά πεδία της Υπολογιστικής Όρασης και του Artificial Intelligence (AI). Αυτό γιατί τα μυστικά της αναγνώρισης σιγά σιγά αποκαλύπτονται μέσω της ανάπτυξης αρχιτεκτονικών όπως είναι τα Convolutional Neural Networks (Deep Learning Algorithm), μια πιο εξελιγμένη μορφή των (ANN), η οποία πλαισιώνεται από big data κατάλληλα για την εκπαίδευση αυτών και συστημάτων προηγμένης τεχνολογίας.



Εικόνα 10: Deep Learning

Αναφορικά με τα παραπάνω το deep learning αποτελεί ένα υπό-πεδίο του Machine Learning. Η κύρια ενασχόλησή του αφορά αλγορίθμους εμπνευσμένους από τη δομή και τη λειτουργία του εγκεφάλου που ονομάζεται Artificial Neural Network (ANN). Η διαδικασία αυτή ακολουθεί συγκεκριμένα βήματα. Αρχικά, αναπαριστά τη λειτουργία του εγκεφάλου μας καθώς οι αλγόριθμοι deep learning είναι παρόμοιοι με τη δόμηση του ανθρώπινου νευρικού συστήματος, όπου κάθε νευρώνας συνδέεται με τους άλλους και διαδίδει τις αντίστοιχες πληροφορίες. Το deep learning άρχισε να έχει μεγάλη συμβολή στην Υπολογιστική όραση το 2012, όταν το γκρουπ του Hinton νίκησε το ImageNet Large Scale Visual Recognition Challenge (ILSVRC) με χρήση τη χρήση του deep learning (A. Krizhevsky, L. Sutskever, 2012).

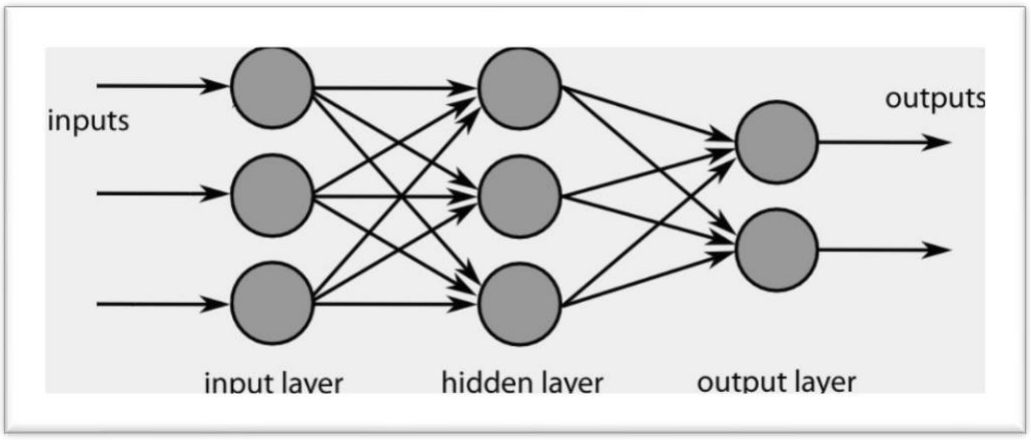
Τα μοντέλα deep learning λειτουργούν σε στρώματα, ένα τυπικό μοντέλο έχει τουλάχιστον 3 στρώματα (layers). Κάθε στρώμα δέχεται πληροφορίες από τα προηγούμενα και διαβιβάζει στα επόμενα

Νευρωνικά δίκτυα για αναγνώριση αντικειμένων

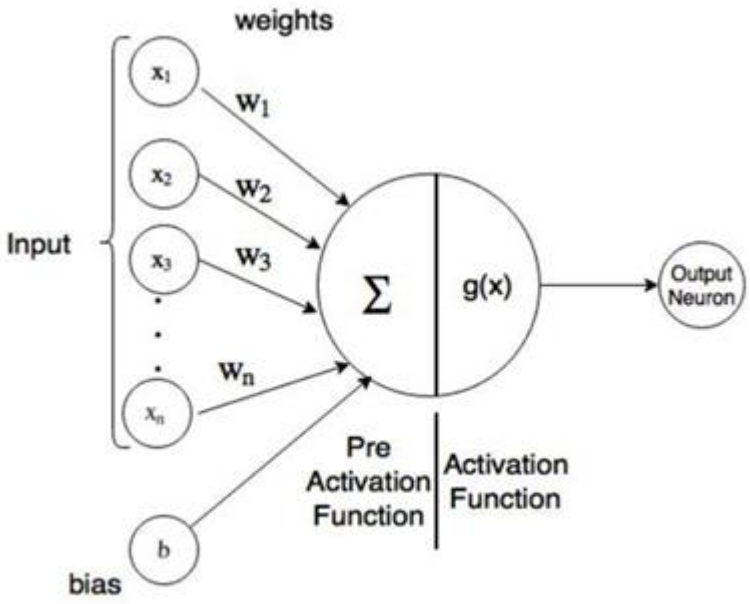
Νευρωνικά δίκτυα πολλαπλών στρωμάτων (feedforward neural networks)

Αυτός ο τύπος νευρωνικού δικτύου είναι ένας από τους απλούστερους τύπους

που χρησιμοποιούνται για αναγνώριση αντικειμένου, όπου οι συνδέσεις μεταξύ των κόμβων δεν σχηματίζουν έναν κύκλο (δεν περιέχει feedback connection, αν επεκταθεί και εισαχθεί αυτή η διαδικασία ονομάζεται recurrent neural network). Αυτό το δίκτυο έχει ένα στρώμα εισόδου (input layer), που συνοδεύεται από κρυφά επίπεδα (hidden layer) και ένα στρώμα εξόδου (output layer). Όλοι οι νευρώνες του στρώματος i συνδέονται με κάθε νευρώνα στο στρώμα i-1. Κάθε μία από αυτές τις συνδέσεις έχει βάρος που καθορίζει την ευαισθησία του νευρώνα στις τιμές εξόδου των νευρώνων του προηγούμενου στρώματος. Κάθε νευρώνας, εξαιρουμένων των νευρώνων στο στρώμα εισόδου, έχει επίσης έναν αριθμό μεροληψίας (bias number), που ενισχύει ή εξασθενεί τη δραστηριότητα του νευρώνα (C. Stergiou and D. Siganos)



Εικόνα 11: simple feedforward neural network



Εικόνα 12: Βασική Μονάδα του νευρωνικού δικτύου

α)Προενεργοποίηση(ενεργοποίηση-εισόδου)

$$a(x) = b + \mathbf{w}^{\mathrm{T}}\mathbf{x}$$

όπου,

a:Λειτουργία-Προ-Ενεργοποίησης

x:διάνυσμα-εισόδου

w:διάνυσμα των βαρών σύνδεσης. Αντιπροσωπεύει τη δύναμη μεταξύ των συνδέσεων.

β: προκατάληψη (bias). Εάν δεν έχουμε εισόδους, το b θα είναι η συνεισφορά μας για τον νευρώνα. Η προκατάληψη δεν είναι παρά μια αρνητική τιμή του threshold. Το threshold σημαίνει την οριακή τιμή που θα έπρεπε να έχει η συνάρτηση ενεργοποίησης σε ένα άλλο κλάδο.

β) Ενεργοποίηση Νευρώνων (Έξοδος): Χρησιμοποιούμε τιμές από τη λειτουργία προ-ενεργοποίησης για να υπολογίσουμε τη λειτουργία ενεργοποίησης.

$$h(x) = g(a(x)) = g(b + \mathbf{w}^{\mathrm{T}}\mathbf{x})$$

h(x): είναι το αποτέλεσμα του νευρώνα

g(x): λειτουργία ενεργοποίησης, υπάρχουν διάφορες λειτουργίες ενεργοποίησης όπως linear, sigmoid, hyperbolic tangent, relu.

Όταν χρησιμοποιείται στην αναγνώριση εικόνας, η είσοδος στο δίκτυο θα είναι η τιμή του pixel για κάθε pixel της εικόνας, έτσι ώστε το στρώμα εισόδου να έχει ένα νευρώνα για κάθε pixel στην εικόνα εισόδου. Οι νευρώνες στο στρώμα εξόδου μπορεί, για παράδειγμα, να κρατήσουν τις τιμές πιθανότητας για κάθε κατηγορία αντικειμένων. Το αντικείμενο με την υψηλότερη τιμή εξόδου επιλέγεται στη συνέχεια ως αποτέλεσμα ανίχνευσης. Για να αποκτηθούν σωστά αποτελέσματα για κάθε εικόνα, το νευρωνικό δίκτυο πρέπει να εκπαιδευτεί (C. M. Bishop, 1995).

Εκπαίδευση του δικτύου σημαίνει ελαχιστοποίηση της συνάρτησης κόστους αυτού του δικτύου. Μία από τις μεθόδους που χρησιμοποιούνται για την ελαχιστοποίηση αυτής της τιμής είναι ο επανα-προσανατολισμός (backpropagation). Αυτή η μέθοδος τροποποιεί τα βάρη των συνδέσεων μεταξύ των στρωμάτων i και i-1 και τις τάσεις των νευρώνων στο στρώμα i-1 με βάση τα σφάλματα στην τιμή του νευρώνα στο στρώμα i (C. Stergiou and D. Siganos). Αυτό σημαίνει ότι το σφάλμα μεταδίδεται προς τα πίσω, μέσω του δικτύου. (Τα Regular Neural Networks (NN) δεν ενδείκνυται να χρησιμοποιηθούν με εικόνες. Θα έπρεπε κάθε pixel να είναι ένας νευρώνας και έτσι θα έπρεπε να υπάρχουν χιλιάδες νευρώνες για την κάθε εικόνα.)

Συνελικτικά Νευρωνικά Δίκτυα (CNN)

Το συνελικτικό νευρωνικό δίκτυο (CNN) είναι ένας ειδικός τύπος νευρωνικού δίκτυού, που έχει σχεδιαστεί για να απαιτεί ελάχιστη προ επεξεργασία (M. Y. W. Teow, 2017). Αποτελείται από πολλαπλά στάδια. Τα μοντέλα τύπου CNN χρησιμοποιούνται για την αναγνώριση εικόνων μετασχηματίζοντας την αρχική εικόνα μέσω layers σε βαθμολογίες κλάσης, κάθε φορά που βλέπει κάτι ενεργοποιείται μια σειρά από layers νευρώνων και κάθε στρώμα θα ανιχνεύσει ένα σύνολο χαρακτηριστικών όπως γραμμές και άκρα. (Y. LeCun, K. Kavukcuoglu, and C. Farabet, 2010).

Υπάρχουν δύο κύριες ενότητες ενός CNN**: η μονάδα μάθησης χαρακτηριστικών (feature learning:conv,relu,pool) και η ταξινόμηση (classification: F.C. και Softmax ή Sigmoid activation function) (M. Y. W. Teow, 2017)**. Η ενότητα μάθησης χαρακτηριστικών μπορεί να περιέχει πολλαπλά στάδια 3 επιπέδων, που αποτελείται από ένα στρώμα συνέλιξης (φίλτρου), ένα στρώμα μη γραμμικότητας (π.χ. RELU) και μια συγκέντρωση χαρακτηριστικών στρώμα (pool) (Y. LeCun, K. Kavukcuoglu, and C. Farabet, 2010).

Το **στρώμα εισαγωγής** είναι μια εικόνα με τις ακόλουθες διαστάσεις (πλάτος (width), ύψος (height), βάθος(depth). Είναι ένας πίνακας από τιμές των pixels.

Στο **επίπεδο συνέλιξης** (φιλτραρίσματος) γίνεται εξαγωγή οπτικών χαρακτηριστικών από μια εικόνα εισόδου. Το συνελικτικό conv() είναι το εσωτερικό γινόμενο της εικόνας εισόδου και του συνελικτικου πυρήνα ή φίλτρου (Kernel, filter ή feature detector), η έξοδος είναι ένας συνελικτικός χάρτης χαρακτηριστικών (feature map) (M. Y. W. Teow, 2017). Στη συνέχεια, μεταφέρουμε το φίλτρο στο επόμενο πεδίο υποδοχής της ίδιας εικόνας εισόδου με συγκεκριμένο βήμα (Stride) και υπολογίζουμε πάλι το εσωτερικό γινόμενο μεταξύ του νέου πεδίου υποδοχής (receptive field) και του ίδιου φίλτρου. Επαναλαμβάνουμε αυτή τη διαδικασία μέχρι να περάσουμε ολόκληρη την εικόνα εισόδου. Η έξοδος πρόκειται να είναι η είσοδος για το επόμενο στρώμα. Παρακάτω παρατίθενται λεπτομερώς βασικοί όροι αυτού του επιπέδου:

- Filter, kernel or Feature detector: είναι ένας πίνακας χρησιμοποιείται για την ανίχνευση χαρακτηριστικών. Ένα τυπικό φίλτρο στο πρώτο στρώμα ενός ConvNet μπορεί να έχει μέγεθος [5x5x3].

- Convolved Feature, Activation Map, Feature Map: είναι το αποτέλεσμα της εξόδου που σχηματίζεται μετατοπίζοντας το φίλτρο πάνω από την εικόνα και υπολογίζοντας το εσωτερικό γινόμενο.

- Receptive Field: είναι μια περιοχή του όγκου της εισόδου που έχει το ίδιο μέγεθος με το φίλτρο.

- Depth: είναι ο αριθμός των φίλτρων.

- Depth column (or fibre): είναι το σύνολο των νευρώνων που δείχνουν όλοι στο ίδιο receptive field.

- Stride: είναι το βήμα μετατόπισης του φίλτρου, πιο κοινό βήμα είναι το 1 και το 2.

- Zero-padding: είναι η διαδικασία που προσθέτει μηδενικά γύρω από τον όγκο της εισόδου, έτσι ώστε οι συνελίξεις να καταλήγουν με τον ίδιο αριθμό εξόδων με τις εισόδους. Εάν δεν χρησιμοποιήσουμε το padding, οι πληροφορίες στα σύνορα θα χαθούν μετά από κάθε στρώμα Conv.

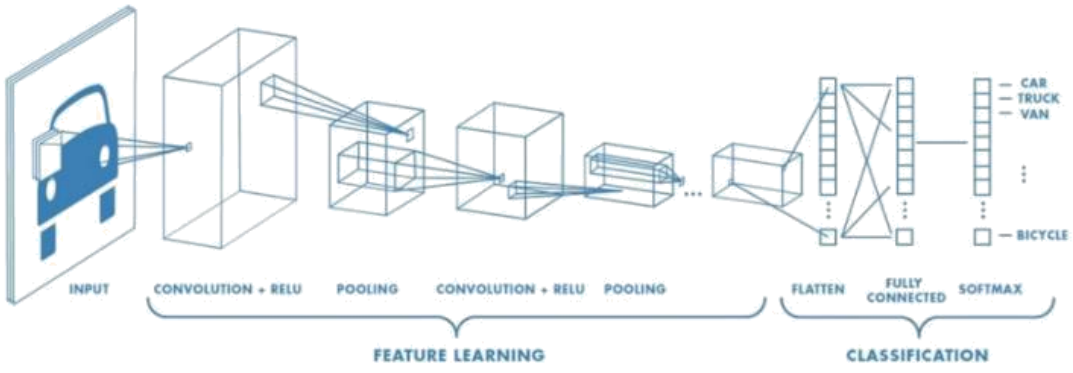
- Parameters Sharing (shared weights): γιατί εάν μια λειτουργία είναι χρήσιμη, θα είναι επίσης χρήσιμο να την αναζητήσετε παντού στην εικόνα.

- Dilation (διαστολή): είναι μια νέα hyperparameter που εισάγεται στο στρώμα της συνέλιξης, είναι στην ουσία φίλτρα με κενά μεταξύ των κελίων.

Το στρώμα **μη γραμμικότητας (activation)** αποτελείται παραδοσιακά από μια λειτουργία σιγμοειδούς ή υπερβολικής εφαπτομένης (Y. LeCun, K. Kavukcuoglu, and C. Farabet, 2010). Το πιο συνηθισμένο είναι το ReLU Layer, το οποίο εφαρμόζει μια συνάρτηση ενεργοποίησης max(0, x), η οποία μετατρέπει τις αρνητικές τιμές σε μηδενικά. Σε αυτό το στρώμα δεν υπάρχουν υπερ-παραμετρικές (hyperparameters).

Το **στρώμα συγκέντρωσης χαρακτηριστικών (pooling layer)** υποδεικνύει τον χάρτη χαρακτηριστικών για τη μείωση του χώρου των διαστάσεων, παράγοντας έτσι μια πιο συμπαγή αναπαράσταση χαρακτηριστικών (M. Y. W. Teow, 2017). Παράλληλα, ελέγχει την υπερφόρτωση και λειτουργεί ανεξάρτητα σε κάθε μέρος (slice) της εισόδου. Υπάρχουν διαφορετικές λειτουργίες όπως η συγκέντρωση Max, η μέση συγκέντρωση ή η συγκέντρωση L2-norm. Ωστόσο, η συγκέντρωση Max είναι ο πιο χρησιμοποιημένος τύπος. Το στρώμα της συγκέντρωσης δεν έχει παραμέτρους (τα βάρη και τις προκαταλήψεις των νευρώνων) και δεν έχει μηδενική συμπλήρωση (zero padding), αλλά έχει δύο υπερ παραμέτρους: το Φίλτρο και Stride (βήμα). Σε πολλούς ανθρώπους δεν τους αρέσει να χρησιμοποιούν ένα στρώμα συγκέντρωσης επειδή απομακρύνουν τις πληροφορίες και το αντικαθιστούν με ένα στρώμα Conv με αυξανόμενο βήμα.

Τέλος υπάρχουν τα **πλήρως συνδεδεμένα στρώματα (Fully Connected layer-FC)**, τα οποία συνδέουν κάθε νευρώνα σε ένα στρώμα με κάθε νευρώνα σε ένα άλλο στρώμα. Το τελευταίο πλήρως συνδεδεμένο επίπεδο χρησιμοποιεί μια λειτουργία ενεργοποίησης softmax ή sigmoid για την ταξινόμηση των παραγόμενων χαρακτηριστικών της εικόνας εισόδου σε διάφορες κατηγορίες με βάση το σύνολο δεδομένων εκπαίδευσης.



Εικόνα 13: architecture of CNN

Όπως φαίνεται στην προηγούμενη εικόνα, υπάρχουν τέσσερις κύριες λειτουργίες στο CNN convolution, το Non Linearity (ReLU), το Pooling, το Classification (Fully Connected Layers)

Τα CNN πλέον είναι ο πρώτος τρόπος που χρησιμοποιείται για εικόνες. Τα συνελικτικά νευρωνικά δίκτυα είναι μια από τις πιο αποτελεσματικές, στον τομέα της υπολογιστικής όρασης, τεχνικές αναγνώρισης. Παράλληλα, έγιναν μια σημαντική τεχνική για τους προγραμματιστές στο σχεδιασμό πιο αποτελεσματικών και εξελιγμένων οπτικών εφαρμογών αναγνώρισης (Y. LeCun, K. Kavukcuoglu, and C. Farabet, 2010).

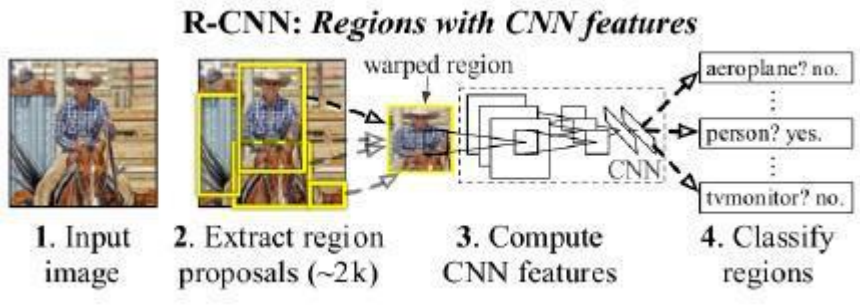
4 *Υψηλής τεχνολογίας ανιχνευτές CNN*

4.1 Γρηγορότερα R-CNN

Η εξελικτική πορεία του συγκεκριμένου μοντέλου ήταν η εξής: η αρχική μορφή ήταν το R-CNN, η οποία εξελίχθηκε στο γρήγορο R-CNN, το οποίο έδωσε τη θέση του στο ακόμη γρηγορότερο R-CNN.

Αναφορικά με το δίκτυο R-CNN, αυτό επηρεάστηκε από το AlexNet (cnn με 8 layer, τα πρώτα 5 είναι conv layer, ακολουθούν 3 F.C layer και κάνει χρήση ReLU), δηλαδή χρησιμοποιεί μέρη αυτού με παραλλαγές καθώς έχει βελτιωθεί για την αναγνώριση πολλών αντικειμένων σε μια εικόνα και όχι μόνο της κατηγοριοποίησης όλης της εικόνας. Αυτό το πετυχαίνει με την διαδικασία Selective Search (Jasper RR Uijlings, 2013). Με αυτό τον τρόπο βάζει το αντικείμενο αναγνώρισης σε ένα ορθογώνιο και η αναγνώριση των αντικειμένων γίνεται από μια βελτιωμένη έκδοση του AlexNet. Στο τελευταίο στάδιο υπάρχει ένα SVM το οποίο κατηγοριοποιεί το αντικείμενο σε κάποια κλάση και το νευρωνικό λαμβάνει σαν εισόδους τις περιοχές που ανήκουν τα αντικείμενα και τις επιστρέφει βελτιωμένες (σαν πρόβλημα απλής παλινδρόμησης).

Το R-CNN δεν μπορεί να χρησιμοποιηθεί για real time αναγνώριση επειδή δεσμεύει πολλή μνήμη και είναι πολύ χρονοβόρο. Οι διάδοχοι του έχουν βελτιωθεί για να μπορούν να χρησιμοποιούν από ενσωματωμένα συστήματα για αναγνώριση σε πραγματικό χρόνο.



Εικόνα 14: R-CNN: Regions with CNN features

Στην εικόνα 16 βλέπουμε:

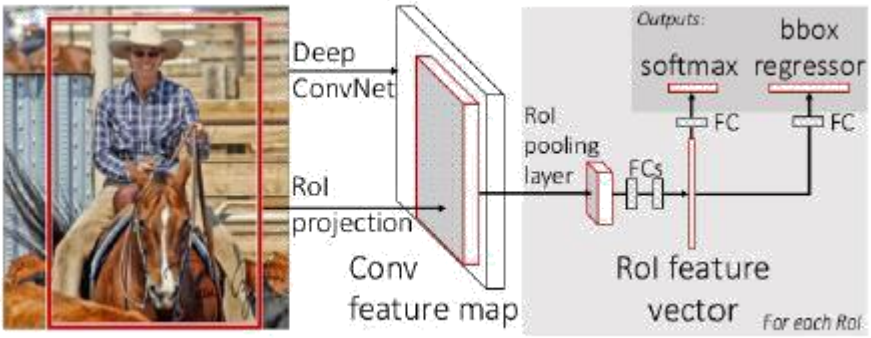
1. Εικόνα εισαγωγής

2.Εξάγει το σύστημα περίπου 2000 περιβλήματα

3.Χρησιμοποιεί ένα συνελικτικό νευρωνικό δίκτυο για να υπολογίσει τα feature για κάθε ένα περίβλημα

4. Κατηγοριοποιεί κάθε περιοχή εντός περιβλήματος με χρήση SVM.

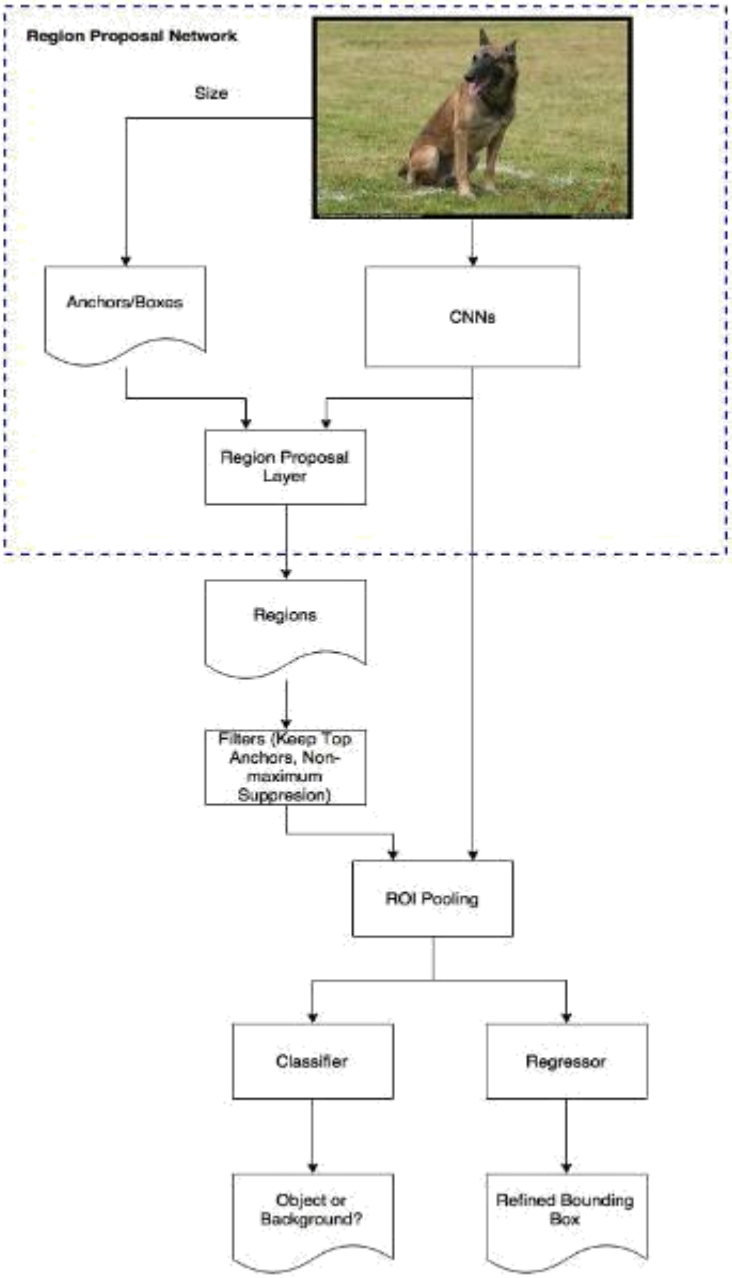
Το **Fast R-CNN** υλοποιήθηκε για να αντιμετωπίσει τα προβλήματα που καθιστούν το R-CNN αργό. Αρχικά εκτελείται το εμπρόσθιο πέρασμα του CNN σε όλη την εικόνα και ύστερα γίνεται pooling σε κάθε πιθανό περίγραμμα αντικειμένου (RoIPool-Region of Interest Pooling), δηλαδή από 2000 περίπου εμπρόσθια περάσματα που έκανε το R-CNN (επειδή το R-CNN κάνει εμπρόσθιο πέρασμα του Alexnet για κάθε πιθανό ορθογώνιο περιβλήματος από την διαδικασία Selective Search) τώρα κάνει ένα στο fast R-CNN. Επιπλέον,, η παλινδρόμηση, για την βελτίωση των περιγραμμάτων των ορθογώνιών, γίνεται παράλληλα με την κατηγοριοποίηση η οποία γίνεται το Softmax classifier που έχει σαν είσοδο το αποτέλεσμα του RoIPool (ενώ στο R-CNN με SVM γινόταν η κατηγοριοποίηση), δηλαδή βάζει και τα τρία μοντέλα να εκτελούνται μαζί σε ένα κοινό δίκτυο και δεν εκπαιδεύει τα τρία μοντέλα ξεχωριστά (CNN για εύρεση χαρακτηριστικών, το SVM, και το μοντέλο απλής παλινδρόμησης για την βελτίωση των περιβλημάτων) όπως γινόταν στο R-CNN. Αυτό έχει σαν αποτέλεσμα η εκπαίδευση του δικτύου να γίνεται αρκετά πιο γρήγορα (2,7 πιο γρήγορα), να αυξάνεται λίγο η ακρίβεια. Ωστόσο, οι χρόνοι εκτέλεσης δεν είναι ακόμα κατάλληλοι για ενσωματωμένα συστήματα.



Εικόνα 15: Fast R-CNN

Στην εικόνα 17 βλέπουμε : Την αρχική εικόνα και τις πολλαπλές περιοχές ενδιαφέροντος (RoI), τα οποία εισάγονται σε ένα συνελικτικό δίκτυο. Κάθε RoI συλλέγεται σε ένα feature map και αντιστοιχίζεται σε feature vector, χρησιμοποιώντας ολικά συνδεμένα επίπεδα. Το δίκτυο έχει δύο διανύσματα χαρακτηριστικών σε κάθε RoI. Ένα που προέρχεται από την έξοδο του Softmax και ένα από την γραμμική παλινδρόμηση. Τέλος, όλα τα κομμάτια του εκπαιδεύονται μαζί Το γρηγορότερο νευρωνικό δίκτυο (Faster R-CNN) που βασίζεται στην περιοχή (region-based) είναι ένας από τους πλέον ακριβείς υπερσύγχρονους ανιχνευτές που είναι διαθέσιμοι σήμερα (S. Ren, K. He, R. Girshick, and J. Sun, 2015). Αυτός ο αλγόριθμος χρησιμοποιεί έναν αλγόριθμο προτάσεων περιοχής για να διατυπώσει υποθέσεις σχετικά με τις τοποθεσίες αντικειμένων. Το συγκεκριμένο δίκτυο αναβάθμισε την προηγούμενη έκδοση, που ονομάζεται Fast R-CNN (R. Girshick, 2015), συνδυάζοντάς την με τα Περιφερειακά Προγράμματα, τα οποία υλοποιούνται σε μια Μονάδα επεξεργασίας γραφικών (GPU) και μοιράζονται συνελικτικά στρώματα με τον ανιχνευτή (S. Ren, K. He, R. Girshick, and J. Sun, 2015). Η καινοτομία του faster R-CNN σε σχέση με τον προκάτοχο του είναι η χρήση ενός νευρωνικού δικτύου RPN (Region Proposal Network), που αντικαθιστά το Selective Search και αυτό έχει σαν αποτέλεσμα ο χρόνος παραγωγής προτάσεων περιοχής να μειωθεί αρκετά. Αυτό εμπεριέχει το ολικά συνδεδεμένο επίπεδο που προτείνει περιοχές, καθώς και τον ανιχνευτή του fast R-CNN που αξιοποιεί τις προτεινόμενες περιοχές από την προηγούμενη διαδικασία.

Συνοπτικά το R-CNN έχει δύο δίκτυα: δίκτυο πρότασης περιοχής (RPN) για την παραγωγή προτάσεων περιοχής και ένα δίκτυο χρησιμοποιώντας αυτές τις προτάσεις για την ανίχνευση αντικειμένων. Το RPN κατατάσσει τα κουτιά περιοχής (anchors) και προτείνει εκείνα που πιθανότατα περιέχουν αντικείμενα.

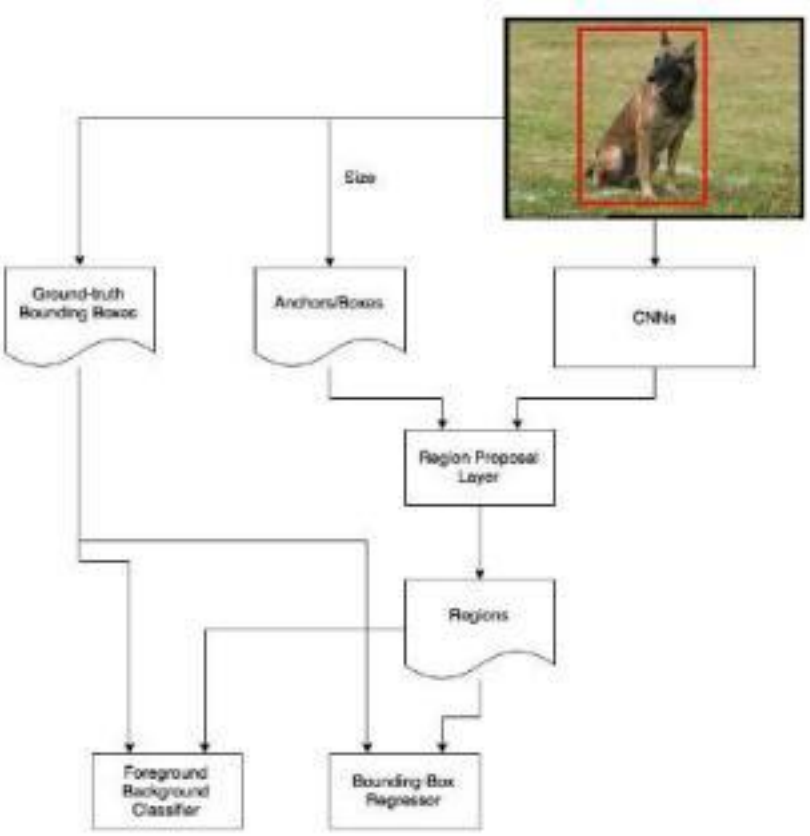


Εικόνα 16: Faster R-CNN

Παρακάτω παρατίθενται λεπτομερώς βασικοί όροι του faster R-CNN:

Anchors έχουν σημαντικό ρόλο στο Faster R-CNN. Ένα anchor είναι ένα κουτί περιοχής. Τα κουτιά-anchor είναι ένα σύνολο προκαθορισμένων κουτιών διαφόρων μεγεθών που χρησιμοποιούνται για την αναγνώριση αντικειμένων εντός του τμήματος πρότασης περιοχών ενδιαφέροντος. (στο default faster R-CNN είναι 9 για κάθε θέση στην εικόνα). Με το RPN μειώνουμε σημαντικά τον αριθμό αυτών των κουτιών.

Region Proposal Network (RPN). Η έξοδος ενός δικτύου πρότασης περιοχής (RPN) είναι μια δέσμη κουτιών (προτάσεων) που θα εξεταστούν από έναν classifier και έναν regressor για να ελέγξουν τελικά την εμφάνιση αντικειμένων. Πιο συγκριμένα, το RPN προβλέπει τη δυνατότητα ενός anchor να είναι υπόβαθρο ή προσκήνιο.



Εικόνα 17: Region Proposal Network In Training

The Classifier of Background and Foreground. Το πρώτο βήμα της εκπαίδευσης ενός classifier είναι η δημιουργία ενός συνόλου εκπαίδευσης. Τα δεδομένα εκπαίδευσης είναι τα anchors που λαμβάνουμε από την παραπάνω διαδικασία και τα κουτία αλήθειας (ground-truth: αναφέρεται στην ακρίβεια της ταξινόμησης, αναφέρεται στην ετικέτα για κάθε δείγμα εκπαίδευσης που γνωρίζουμε, δηλαδή ξέρουμε ποια κατηγορία ανήκει σε κάθε δείγμα εκπαίδευσης). Το πρόβλημα που πρέπει να λύσουμε εδώ είναι πώς χρησιμοποιούμε τα κιβώτια της αλήθειας για να επισημάνουμε (label) τα anchors. Η βασική ιδέα εδώ είναι ότι θέλουμε να επισημάνουμε ότι τα anchors που έχουν τις υψηλότερες αλληλεπικαλύψεις με κουτιά αλήθειας είναι το προσκήνιο, ενώ εκείνα με χαμηλότερες επικαλύψεις είναι το φόντο. Κάθε θέση στον χάρτη χαρακτηριστικών (feature map) διαθέτει 9 anchors και κάθε anchor έχει δύο πιθανές ετικέτες (φόντο, προσκήνιο). Εάν κάνουμε το βάθος του χάρτη χαρακτηριστικών ως 18 (9 άγκυρες x 2 ετικέτες), θα κάνουμε κάθε anchor να έχει ένα διάνυσμα με δύο τιμές (logit) που αντιπροσωπεύει το προσκήνιο και το φόντο. Εάν τροφοδοτούμε το logit σε λειτουργία ενεργοποίησης softmax / logistic regression, θα προβλέψει τις ετικέτες. Τώρα τα δεδομένα εκπαίδευσης είναι πλήρη με χαρακτηριστικά και ετικέτες.

The Regressor of Bounding Box. Μετά το label των anchors μπορούμε επίσης να διαλέξουμε τα anchors βάσει των παρόμοιων κριτηρίων για τον regressor για βελτιωθεί το πλαίσιο του κουτιού.

Η **συνολική απώλεια του RPN** είναι ένας συνδυασμός της απώλειας ταξινόμησης και της απώλειας παλινδρόμησης.

ROI Pooling. Μετά το RPN, προτείνονται περιοχές με διαφορετικά μεγέθη. Οι περιοχές διαφορετικού μεγέθους σημαίνουν διαφορετικούς χάρτες χαρακτηριστικών CNN. Δεν είναι εύκολο να δημιουργήσουμε μια αποτελεσματική δομή για να εργαστούμε σε λειτουργίες με διαφορετικά μεγέθη. Η διαδικασία ROI Pooling μπορεί να απλοποιήσει το πρόβλημα μειώνοντας τους χάρτες χαρακτηριστικών στο ίδιο μέγεθος. Σε αντίθεση με το Max-Pooling που έχει ένα σταθερό μέγεθος, το ROI Pooling χωρίζει τον χάρτη χαρακτηριστικών εισόδου σε έναν σταθερό αριθμό περίπου ίσων περιοχών και στη συνέχεια εφαρμόζουμε το Max-Pooling σε κάθε περιοχή.

TRAINING. Μπορεί να γίνει με δύο τρόπους:

- να εκπαιδεύσει εναλλακτικά το RPN, και τον τελικό ταξινομητή και regressor.
- να τους εκπαιδεύσει ταυτόχρονα από κοινού.

Σε αυτή την διπλωματική θα γίνει εφαρμογή του Faster R-CNN.

4.2 YOLO (You Only Look Once)

Το σύστημα ανίχνευσης YOLO είναι ένας οπτικός ανιχνευτής, ο οποίος μπορεί να αναγνωρίσει πολλαπλές κατηγορίες αντικειμένων σε μια εικόνα. Σε αντίθεση με άλλους, παρόμοιους ανιχνευτές, το YOLO χρησιμοποιεί μόνο ένα συνελικτικό νευρωνικό δίκτυο για την πρόβλεψη όλων των πλαισίων οριοθέτησης και των πιθανοτήτων για κάθε κατηγορία αντικειμένου. Όλες αυτές οι προβλέψεις είναι ταυτόχρονες, με αποτέλεσμα οι διεργασίες YOLO να δίνουν εικόνα μόνο σε ένα πέρασμα. Χάρη σε αυτή την ικανότητα, το YOLO είναι γρήγορο στην ανίχνευση αντικειμένων και μπορεί να χρησιμοποιηθεί σε πραγματικό χρόνο (J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 2016).

Σε αντίθεση με τις τεχνικές που βασίζονται σε προτάσεις συρόμενων παραθύρων και περιοχών, το YOLO βλέπει ολόκληρη την εικόνα κατά τη διάρκεια της εκπαίδευσης και του χρόνου δοκιμής και ως εκ τούτου κωδικοποιεί πληροφορίες σχετικές με τις τάξεις καθώς και την εμφάνισή τους (J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 2016).

Το σύστημα λειτουργεί ως εξής. Πριν από την ανίχνευση, η εικόνα εισόδου κλιμακώνεται στα 448 x 448 pixels. Στη συνέχεια διαιρείται σε δίκτυο S x S. Το κελί που περιλαμβάνει το κέντρο ενός αντικειμένου είναι υπεύθυνο για την ανίχνευση αυτού του αντικειμένου. Στη

συνέχεια, κάθε κελί κάνει προβλέψεις n πλαισίων και υπολογίζει το βαθμό εμπιστοσύνης για κάθε μία οριοθέτηση. Αυτές οι βαθμολογίες αξιοπιστίας αντανακλούν την εμπιστοσύνη του μοντέλου, όπου το κουτί περιέχει ένα αντικείμενο καθώς και την ακρίβεια αυτού του κουτιού (J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 2016). Το δίκτυο ανίχνευσης YOLO έχει 24 συνελικτικά στρώματα, ακολουθούμενα από 2 πλήρως συνδεδεμένα στρώματα. Το Fast YOLO έχει μόνο 9 στρώματα συνέλιξης αντί για 24. Αυτή είναι η μόνη διαφορά μεταξύ αυτών των δύο εκδόσεων (J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, 2016).

Μια νεότερη έκδοση, YOLOv2, έχει πολλά πλεονεκτήματα σε σύγκριση με την παλαιότερη εκδοχή. Το πρώτο πλεονέκτημα αφορά τη δυνατότητα τροποποίησης της ανάλυσης της εισόδου της εικόνας με αποτέλεσμα το συντονισμό της ακρίβειας και της ταχύτητας του ανιχνευτή (J. Redmon and A. Farhadi, 2017). Η δομή του δικτύου έχει επίσης αλλάξει. Το YOLOv2 έχει 19 συνελικτικά στρώματα και 5 στρώματα maxpool (J. Redmon and A. Farhadi, 2017)**.**

Το YOLOv3 χρησιμοποιεί ένα νέο δίκτυο για εξαγωγή χαρακτηριστικών. Αυτό το δίκτυο έχει 53 συνελικτικά στρώματα (Darknet-53). Η δομή του δικτύου μπορεί να παρατηρηθεί στο (Redmon and A. Farhadi, 2018). Αυτός ο ανιχνευτής έχει την ίδια ακρίβεια με το SSD, αλλά είναι τρεις φορές γρηγορότερος (Redmon and A. Farhadi, 2018).

4.3 Ανιχνευτής πολλαπλών θυρών με μία λήψη (SSD)

Ο ανιχνευτής πολλαπλών θυρίδων (SSD) μιας λήψης χρησιμοποιεί παρόμοια προσέγγιση με το YOLO. Ωστόσο δεν χρησιμοποιεί μεθόδους πρόβλεψης πλαισίου οριοθέτησης, γεγονός που βελτιώνει σημαντικά την ταχύτητα. Αντίθετα, ο αλγόριθμος χρησιμοποιεί μικρό συνέλιγμα φίλτρου για την πρόβλεψη της κατηγορίας του αντικειμένου και τις αντισταθμίσεις σε θέση με το πλαίσιο οριοθέτησης, χρησιμοποιώντας ξεχωριστά φίλτρα για διαφορετικές ανιχνεύσεις αναλογίας διαστάσεων και εφαρμόζει αυτά τα φίλτρα σε πολλαπλά feature maps από τα επόμενα στρώματα του δικτύου για να εκτελέσει ανίχνευση σε πολλαπλές κλίμακες. Με αυτές τις τροποποιήσεις, μπορεί να επιτευχθεί υψηλή ακρίβεια χρησιμοποιώντας σχετικά χαμηλή ανάλυση εισόδου, αυξάνοντας όμως συνεχώς την ταχύτητα του ανιχνευτή. Το SSD πρόγραμμα, είναι ταχύτερο και έχει καλύτερη ακρίβεια από το αρχικό YOLO (Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C, 2016).

5 *Εργαλεία που χρησιμοποιηθήκαν*

5.1 Python

Η Python[3] είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου που δημιουργήθηκε από τον Guido Van Rossum, το 1991. Είναι σχεδιασμένη για να είναι ευανάγνωστη για προγραμματισμό γενικού σκοπού. Η Python διαθέτει ένα τεράστιο σύνολο βιβλιοθηκών μπορεί να χρησιμοποιηθεί για μηχανική μάθηση και προγραμματισμό τεχνητής νοημοσύνης (π.χ. NumPy, Tensorflow, Keras). Πολλές από αυτές τις βιβλιοθήκες χρησιμοποιούν C ή Fortran ως backend για να φέρουν εις πέρας βαρείες υπολογιστικές διεργασίες πολύ γρήγορα.

5.2 OpenCV

Το OpenCV[4] (Open Source Computer Vision Library) είναι μια πλατφόρμα ανοιχτού κώδικα και βιβλιοθήκη λειτουργιών προγραμματισμού που στοχεύει κυρίως στην υπολογιστική όραση. Έχει διασυνδέσεις C ++, Python και Java και έχει σχεδιαστεί αποτελεσματικά για υπολογιστικές εφαρμογές. Το OpenCV περιέχει πολλές χρήσιμες λειτουργίες και υλοποιήσεις αλγορίθμων. Κάποιοι από αυτούς είναι:

- Είσοδος ροής βίντεο / φωτογραφίας.
- Ταξινομητής Haar-cascade.
- Παρακολούθηση αντικειμένων.
- Εργαλεία σχεδίασης.
- Κοπή εικόνας.

3. https://www.python.org
4. https://opencv.org

5.3 TensorFlow

Το TensorFlow[5] είναι ένα πλαίσιο ανοιχτού κώδικα για υψηλή απόδοση αριθμητικού υπολογισμού. Η ευέλικτη αρχιτεκτονική του επιτρέπει την εύκολη ανάπτυξη υπολογισμού σε διάφορες πλατφόρμες (GPU, CPU, TPU) και από επιτραπέζιους υπολογιστές σε συστοιχίες (clusters) διακομιστών σε κινητές συσκευές και συσκευές άκρων. Αναπτύχθηκε από ερευνητές και μηχανικούς από την ομάδα του Google Brain μέσω του AI της Google. Το πλαίσιο TensorFlow παρέχει ισχυρή υποστήριξη για την μηχανική μάθηση, τη βαθιά εκμάθηση και τον ευέλικτο αριθμητικό υπολογισμό. Το TensorFlow είναι εξαιρετική επιλογή για τη δημιουργία βαθέων Νευρωνικών Δικτύων (Deep Neural Network) με ατελείωτες δυνατότητες για ακριβής προσαρμογή κάθε μεμονωμένου στοιχείου του δικτύου (Abadi, M.,Agarwal, A., 2015).

5.4 Cuda

Το CUDA[6] είναι μια παράλληλη υπολογιστική πλατφόρμα και μοντέλο προγραμματισμού που αναπτύχθηκε από τη NVIDIA για υπολογιστικές λειτουργίες σε γραφικές μονάδες επεξεργασίας (GPUs). Με την CUDA, οι προγραμματιστές είναι σε θέση να επιταχύνουν τις υπολογιστικές εφαρμογές αξιοποιώντας τη δύναμη της GPU.

Στις εφαρμογές που έχουν επιταχυνθεί με GPU, το διαδοχικό τμήμα του φόρτου εργασίας τρέχει στη CPU - η οποία είναι βελτιστοποιημένη για την απόδοση σε ένα λογικό πυρήνα - ενώ το τμήμα που καταναλώνει μεγάλο μέρος της εφαρμογής τρέχει σε χιλιάδες πυρήνες GPU παράλληλα.

Nvidia Cudnn

Η βιβλιοθήκη NVIDIA CUDA Deep Neural Network-cuDNN[7] είναι μια βιβλιοθήκη επιταχυνόμενων υπολογισμών με GPU για βαθιά νευρωνικά δίκτυα. Το cuDNN παρέχει εξαιρετικά συντονισμένες εφαρμογές για τυπικές ρουτίνες, όπως στρώματα (layers) εμπρόσθιας και προς τα πίσω συνέλιξης, συγκέντρωσης, κανονικοποίησης και ενεργοποίησης.

Η βαθιά εκμάθηση των ερευνητικών και των προγραμματιστών πλαισίων παγκοσμίως βασίζονται στο cuDNN για επιτάχυνση GPU υψηλής απόδοσης. Τους επιτρέπει να επικεντρωθούν στην εκπαίδευση νευρωνικών δικτύων και στην ανάπτυξη εφαρμογών λογισμικού αντί να ξοδεύουν χρόνο σε συντονισμό επιδόσεων GPU χαμηλού επιπέδου. Το cuDNN επιταχύνει τα ευρέως χρησιμοποιούμενα πλαίσια βαθιάς μάθησης, όπως τα Caffe, Keras, MxNet, TensorFlow και PyTorch.

5.5 Matlab

Η MATLAB[8] (matrix laboratory) είναι ένα περιβάλλον αριθμητικής υπολογιστικής και μια προγραμματιστική γλώσσα τέταρτης γενιάς. Η οποία δημιουργήθηκε από τον Cleve Moler, πρόεδρος του τμήματος πληροφορικής του νέου Μεξικού. Αποθηκεύει και κάνει τις πράξεις με βάση την άλγεβρα μητρών. Χρησιμοποιείται κατά κύριο λόγο για την επίλυση μαθηματικών προβλημάτων, επίσης είναι πολύ "ισχυρό" και μπορεί να χρησιμοποιηθεί και για προγραμματισμό καθώς περιέχει εντολές από την C++. Στη παρούσα εργασία χρησιμοποιήθηκε για την μετατροπή .mha αρχείων (3D εικόνων - MRI) σε αρχεία .jpg ,με τα οποία θα εκπαιδευτεί το νευρωνικό μας δίκτυο.

5. https://www.tensorflow.org
6. https://developer.nvidia.com/cuda-zone
7. https://developer.nvidia.com/cudnn
8. https://www.mathworks.com/products/matlab.html

6 *Εφαρμογή στο Faster-RCNN για αναγνώριση όγκου στον εγκέφαλο*

Χαρακτηριστικά του υπολογιστή που έγινε η εκπαίδευση :

GPU: GeForce MX110 2GB

CPU: Intel i7-8550U 8^{th} Gen

RAM: 8GB

6.1 Περιγραφή

Ο σκοπός αυτής της εργασίας είναι να εκπαιδευτεί ένας ταξινομητής ανίχνευσης αντικειμένων σε νευρωνικό δίκτυο, με σκοπό την αναγνώριση όγκου και πιο συγκεκριμένα εάν είναι καλοήθης (LGG) ή κακοήθης (HGG). Στο τέλος αυτής της διεργασίας, θα δημιουργηθεί ένα μοντέλο που μπορεί να εντοπίσει και να σχεδιάσει ορθογώνια γύρω από τις περιοχές όπου υπάρχει γλοίωμα στον εγκέφαλο σε εικόνες, βίντεο ή σε ροή κάμερας webcam. Η διαδικασία διεκπεραιώνεται σε λειτουργικό σύστημα Windows με χρήση κατάλληλων εργαλείων και frameworks.

6.2 Βήματα εργασίας

Εγκατάσταση των απαραίτητων frameworks

Κάνουμε εγκατάσταση του Tensorflow-GPU 1.8 (εργαλείο που επιτρέπει στον υπολογιστή να χρησιμοποιεί την κάρτα γραφικών για να παρέχει πρόσθετη ισχύ επεξεργασίας κατά την εκπαίδευση), στη συνέχεια του CUDA v10.0 και του cuDNN 7.4, που είναι συμβατά με το tensorflow-gpu (έχουν αναλυθεί σε προηγούμενο κεφάλαιο). Επίσης το project υλοποιείται σε Python 3.6 και Anaconda environment[9].

Κατεβάζουμε τα απαραίτητα repository από το GitHub

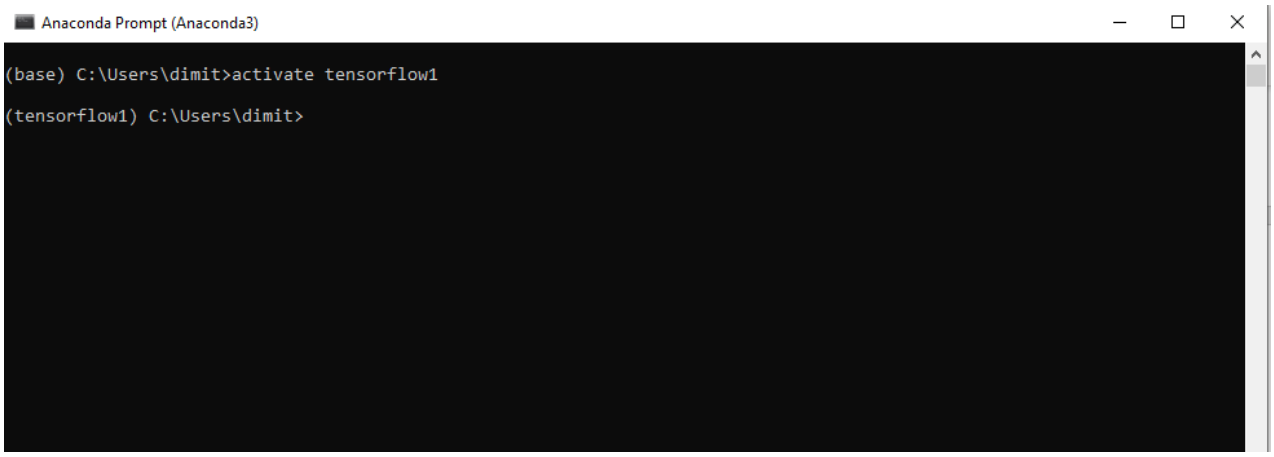
Αρχικά κατεβάζουμε το TensorFlow Object Detection API[10] (το οποίο είναι λογισμικό ανοιχτού κώδικα), που περιέχει συγκεκριμένη δομή καταλόγου και απαιτεί πρόσθετα πακέτα python, καθώς και συγκεκριμένες προσθήκες στα PATH και PYTHONPATH για να μπορέσουμε να τρέξουμε ή να εκπαιδεύσουμε το δικό μας μοντέλο ανίχνευσης αντικειμένων. Έπειτα το αποθηκεύουμε σε ένα φάκελο στον δίσκο του υπολογιστή, όπου αυτός ο φάκελος θα περιέχει όλο το πλαίσιο ανίχνευσης (τις εικόνες ανίχνευσης, τις πληροφορίες εκπαίδευσης, τον εκπαιδευμένο ταξινομητή και όλα τα configuration files) που χρειάζονται για το μοντέλο ανίχνευσης αντικειμένων.

Στη συνέχεια κατεβάζουμε το μοντέλο Faster-RCNN-Inception-V2 από το Tensorflow model zoo[11], το οποίο περιέχει διάφορα προ εκπαιδευμένα μοντέλα ανίχνευσης αντικειμένων δηλαδή προ εκπαιδευμένους ταξινομητές με συγκεκριμένες αρχιτεκτονικές νευρωνικών δικτυών. Έπειτα αποθηκεύουμε αυτό το αρχείο στο φάκελο που έχουμε δημιουργήσει και συγκεκριμένα στη θέση

C:\tensorflow1\models\research\object_detection.

Φτιάχνουμε το εικονικό περιβάλλον για την εφαρμογή

Δημιουργούμε το virtual environment (σαν ένα αυτόνομο περιβάλλον που περιέχει μια έκδοση python και συγκεκριμένα πακέτα της χωρίς να επηρεάζεται από κάποια άλλη εγκατάσταση) από το anaconda prompt. Το Anaconda είναι ένας διαχειριστής πακέτων, ένας διαχειριστής περιβάλλοντος και μια διανομή Python που περιέχει μια συλλογή από πολλά πακέτα ανοιχτού κώδικα, πράγμα που είναι πολύ σημαντικό, καθώς όταν χρειάζεστε πολλά διαφορετικά πακέτα (numpy, scikit-learn, scipy, pandas κ.α), τα οποία προ εγκαθίστανται με την εγκατάσταση του Anaconda. Αυτό είναι εξαιρετικά επωφελές δεδομένου ότι δεν χρειάζεται να διαχειριζόμαστε οι ίδιοι τις εξαρτήσεις μεταξύ πολλαπλών πακέτων για το tensorflow-gpu και το ενεργοποιούμε.



Εικόνα 18: Activate Virtual Environment

Στη συνέχεια κάνουμε εγκατάσταση τα απαραίτητα πακέτα από το anaconda prompt:

- Protobuf (ορίζουμε την δομή και την σειρά των δεδομένων του προγράμματος)
- Pillow (για επεξεργασία εικόνας)
- Lxml (βιβλιοθήκη για υποστήριξη XML αρχείων)
- Cython (βελτιωμένος μεταγλωττιστής για python και επεκτάσεων C, για εύκολη αλληλεπίδραση)
- Matplotlib (βιβλιοθήκη σχεδίασης)
- Pandas (είναι ένα πακέτο που παρέχει γρήγορες δομές δεδομένων σχεδιασμένες να κάνουν την εργασία με "σχεσιακά" δεδομένα πιο απλή)
- Opencv-python (υποστηρίζει πολλούς αλγορίθμους που σχετίζονται με το Computer Vision and Machine Learning και το numpy για αριθμητικές λειτουργίες)

9. *https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)*
10. *https://github.com/tensorflow/models*
11. *https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md*

```
(tf_gpu) C:\> conda install -c anaconda protobuf
(tf_gpu) C:\> pip install pillow
(tf_gpu) C:\> pip install lxml
(tf_gpu) C:\> pip install Cython
(tf_gpu) C:\> pip install jupyter
(tf_gpu) C:\> pip install matplotlib
(tf_gpu) C:\> pip install pandas
(tf_gpu) C:\> pip install opencv-python
```

Εικόνα 19: Εγκατάσταση απαραίτητων πακέτων

Εδώ πρέπει να δημιουργηθεί ένα PYTHONPATH

(Tf_gpu)C:\>setPYTHONPATH=C:\tensorflow1\models;C:\tensorflow1\models\rese
arch;C:\tensorflow1\models\research\slim

6.3 Δημιουργία των protobufs files

Κάνουμε compile τα protobufs (protocol buffers) που είναι μια μέθοδος για serialization δομημένων δεδομένων και εκτελώ το setup.py, τα οποία χρησιμοποιούνται από το tensorflow για την ρύθμιση των παραμέτρων των μοντέλων και της εκπαίδευσης, καθώς περιγράφουμε την δομή των δεδομένων που χρειάζεται το σύστημα σε ένα αρχείο περιγραφής (.proto). Τα αρχεία .proto γίνονται compile με την εντολή protoc και δημιουργεί κώδικα που μπορεί να επικαλείται ένας αποστολέας ή αποδέκτης αυτών των δομών δεδομένων.

Στο Anaconda Command Prompt μπαίνω στο directory \models\research και εκτελώ την εντολή

protoc--python_out=..\object_detection\protos\anchor_generator.proto

.\object_detection\protos\argmax_matcher.proto.\object_detection\protos\bipartite_ma
tcher.proto.\object_detection\protos\box_coder.proto.\object_detection\protos\box_pre
dictor.proto.\object_detection\protos\eval.proto.\object_detection\protos\faster_rcnn.p
roto.\object_detection\protos\faster_rcnn_box_coder.proto.\object_detection\protos\gr
id_anchor_generator.proto.\object_detection\protos\hyperparams.proto.\object_detecti
on\protos\image_resizer.proto.\object_detection\protos\input_reader.proto.\object_det
ection\protos\losses.proto.\object_detection\protos\matcher.proto.\object_detection\pr

otos\mean_stddev_box_coder.proto.\object_detection\protos\model.proto.\object_dete
ction\protos\optimizer.proto.\object_detection\protos\pipeline.proto.\object_detection\
protos\post_processing.proto.\object_detection\protos\preprocessor.proto.\object_dete
ction\protos\region_similarity_calculator.proto.\object_detection\protos\square_box_c
oder.proto.\object_detection\protos\ssd.proto.\object_detection\protos\ssd_anchor_gen
erator.proto.\object_detection\protos\string_int_label_map.proto.\object_detection\pro
tos\train.proto.\object_detection\protos\keypoint_box_coder.proto.\object_detection\p
rotos\multiscale_anchor_generator.proto.\object_detection\protos\graph_rewriter.proto

Τέλος τρέχω :

(tf_gpu) C:\tensorflow1\models\research> python setup.py build

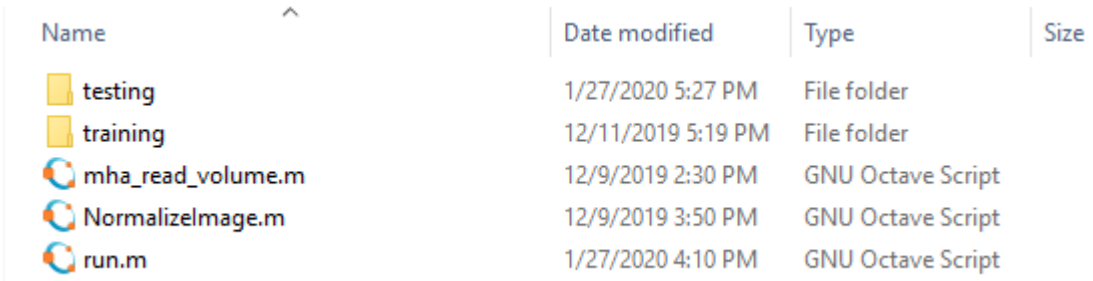
(tf_gpu) C:\tensorflow1\models\research> python setup.py install

6.4 Συλλογή φωτογραφιών και τοποθέτηση ετικετών στις φωτογραφίες

Για τη εκπαίδευση χρησιμοποιήθηκε το σύνολο δεδομένων από τον διαγωνισμό του BRATS2015, από το την βάση δεδομένων ιατρικών αρχείων για ερευνητικούς σκοπούς Smir. Το συγκεκριμένο dataset περιέχει 2 σετ αρχείων, το ένα σετ έχει μαγνητικές ακτινογραφίες (MRI - Magnetic resonance imaging) με αναγνώριση όγκου LGG και το 2ο με αναγνώριση όγκου HGG. Το format των αρχείων και των 2 σετ είναι .mha. Το format .mha είναι μια τρισδιάστατη απεικόνιση (3D) ιατρικών δεδομένων (μαγνητικών τομογραφιών κλπ.), γνωστό ως ITK MetaImage[12]. Άλλες υποστηριζόμενες μορφές αρχείων είναι .raw και .mhd.

Στην συνέχεια, με την βοήθεια της Matlab και την παραμετροποίηση δύο συναρτήσεων (βλ. εικόνα 20) μετατρέψαμε το σύνολο των 2 σετ από τη μορφή .mha σε .jpg. Οι συναρτήσεις mha_read_volume() χρησιμοποιείτε για την ‘ανάγνωση’ των .mha, ενώ η συνάρτηση NormalizeImage() χρησιμοποιείτε για το normalization (μετατροπή) των double μεταβλητών σε unit8. Την συνάρτηση mha_read_volume() την βρήκαμε από το πακέτο Read Medical Data 3D[13] της Matlap, το οποίο είναι ένα free open source code που περιέχει πηγαίους κώδικες για τη ανάγνωση ιατρικών δεδομένων. Τέλος, στην συνάρτηση run() έχουμε φτειάξει μια λίστα στην οποία στοιβάζονται τα αρχεία .mha και μια for() η οποία κάνει προσπέλαση τη λίστα αυτή. Μέσα στην for() καλούμε κατάλληλα τις 2 προαναφερόμενες συναρτήσεις και με την εντολή imwrite() κάνουμε την αποθήκευση των αρχείων από .raw σε .jpg (βλ. εικόνα 21). Όλες οι συναρτήσεις πρέπει να είναι στον φάκελο όπου βρίσκονται τα αρχεία (.mha) .

12. https://itk.org/Wiki/ITK/File_Formats#File_Formats_and_Pixel_Types



Name	Date modified	Type	Size
testing	1/27/2020 5:27 PM	File folder	
training	12/11/2019 5:19 PM	File folder	
mha_read_volume.m	12/9/2019 2:30 PM	GNU Octave Script	
NormalizeImage.m	12/9/2019 3:50 PM	GNU Octave Script	
run.m	1/27/2020 4:10 PM	GNU Octave Script	

Εικόνα 20: Functions

mha_read_header.m | NormalizeImage.m | run.m | mha_read_volume.m

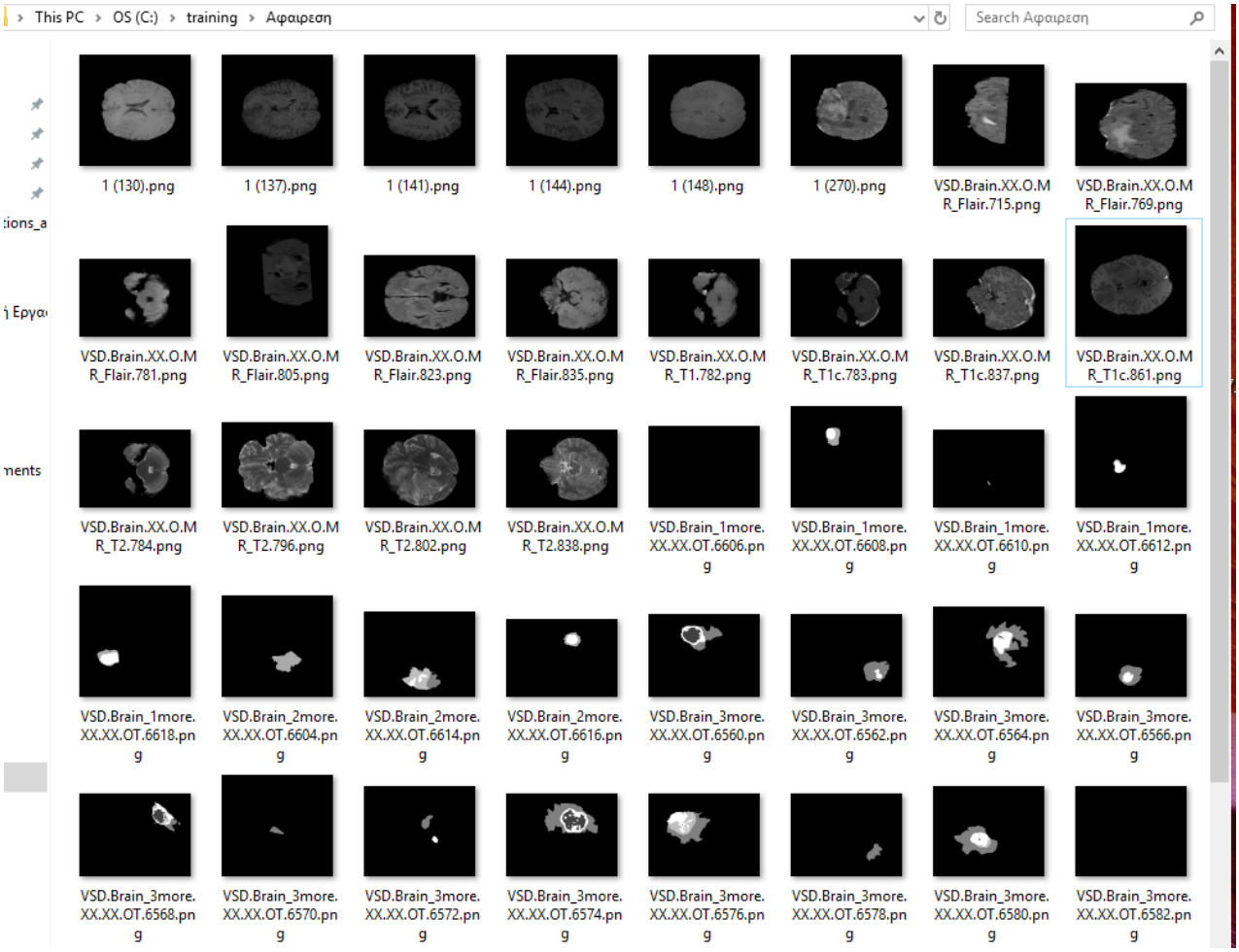
```
function run()
    fileListStruct = dir('*.mha');

    for i = 1: length(fileListStruct)
        sprintf('%d out of %d', i, length(fileListStruct) )
        currentFileName = fileListStruct(i).name;
        currentData = mha_read_volume(currentFileName);
        rawFileName = convertCharsToStrings(extractBetween( currentFileName, 1, ( strlength(currentFileName) - 4 ) ));
        newCurrentData = NormalizeImage( double(currentData) );
        imwrite(squeeze(newCurrentData(:,:,round(end/2))), strcat(rawFileName, '.jpg') );
    end
end
```

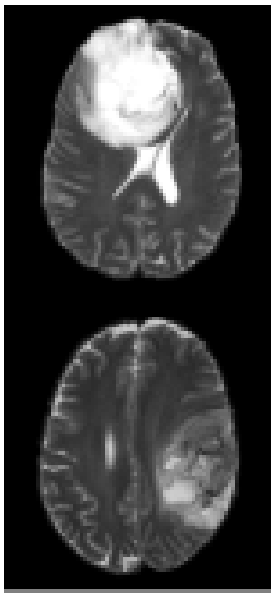
Εικόνα 21: Functions run() - code

Το αρχικό σύνολο των αρχείων στον φάκελο LGG ήταν 270, ενώ στον HGG ήταν 899, μετά την μετατροπή το τελικό σύνολο των αρχείων που έμειναν ήταν 153 και 317 αντίστοιχα, δηλαδή σε άθροισμα 470 εικόνες. Από τις οποίες θα δημιουργήσουμε το train και test data set με μια αναλογία περίπου 26% των εικόνων για το test set και το υπόλοιπο 74% για το train set. Η σημαντική μείωση που σημειώθηκε στο σύνολο εικόνων προήλθε αφαιρώντας χειροκίνητα τις εικόνες που δεν φαινόταν ξεκάθαρα εάν υπήρχε όγκος σε κάποιο σημείο του εγκεφάλου και σε εικόνες που αλλοιώθηκαν σε μεγάλο βαθμό κατά την διάρκεια της μετατροπής (βλ. εικόνα 22).

Εικόνες δειγμάτων των δύο βαθμών γλοιώματος (LGG - HGG) παρουσιάζονται στην εικόνα 23. Μπορεί να παρατηρηθεί από το σχήμα ότι είναι πολύ δύσκολο να γίνει διάκριση σε κάθε περίπτωση, με βάση μόνο τα φαινότυπα που είναι ορατά στο ανθρώπινο μάτι. Ως εκ τούτου, τα χαρακτηριστικά που αντλούνται από το δίκτυο πρότασης περιοχής (RPN) ενός Faster-RCNN αναμένεται να είναι χρήσιμα στην μη διαφοροποίηση μεταξύ τους.



Εικόνα 22: Αφαίρεση εικόνων μετά την μετατροπή



HGG

LGG

Εικόνα 23: Οπτικές διαφορές HGG-LGG

6.5 Τοποθέτηση ετικετών στις φωτογραφίες

Για αυτή την διαδικασία θα χρησιμοποιήσουμε το Labellmg[14] , το οποίο αποτελεί ένα απλό εργαλείο για την δημιουργία label για κάθε εικόνα χειροκίνητα. Ο σχολιασμός του σχήματος αποθηκεύεται σε XML file σε μορφή PASCAL VOC (Visual Object Class), τη μορφή που χρησιμοποιείται από το ImageNet.

13. *https://www.mathworks.com/matlabcentral/fileexchange/29344-read-medical-data-3d?fbclid=IwAR0cxFmenSiq-WLv9cYEeW9L1bP6_-lqokKVaiU8zfWd0CWb_ejA9S49TQA*
14. *https://github.com/tzutalin/labelImg*

Εικόνα 24: Δημιουργία label

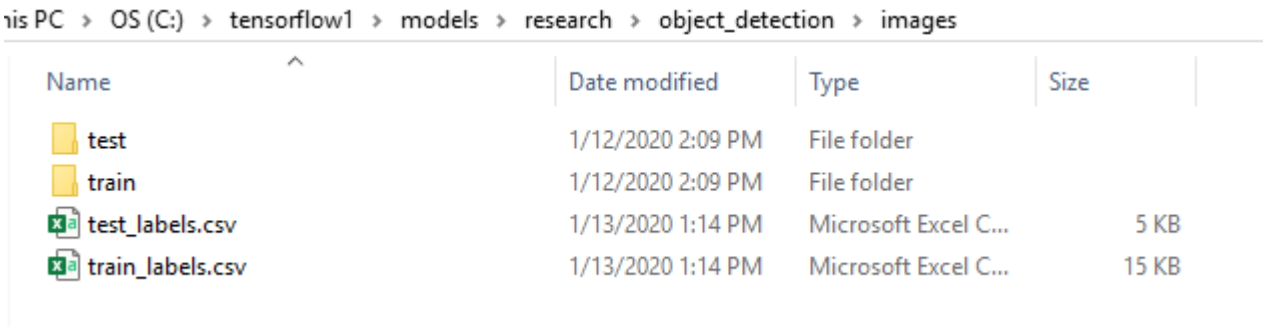
```
<annotation>
    <folder>LGG-noexe(jpg)</folder>
    <filename>image001.jpg</filename>
    <path>C:\training\LGG-noexe(jpg)\image001.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>240</width>
        <height>240</height>
        <depth>1</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>LGG</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>53</xmin>
            <ymin>64</ymin>
            <xmax>150</xmax>
            <ymax>127</ymax>
        </bndbox>
    </object>
</annotation>
```

Εικόνα 25: xml file

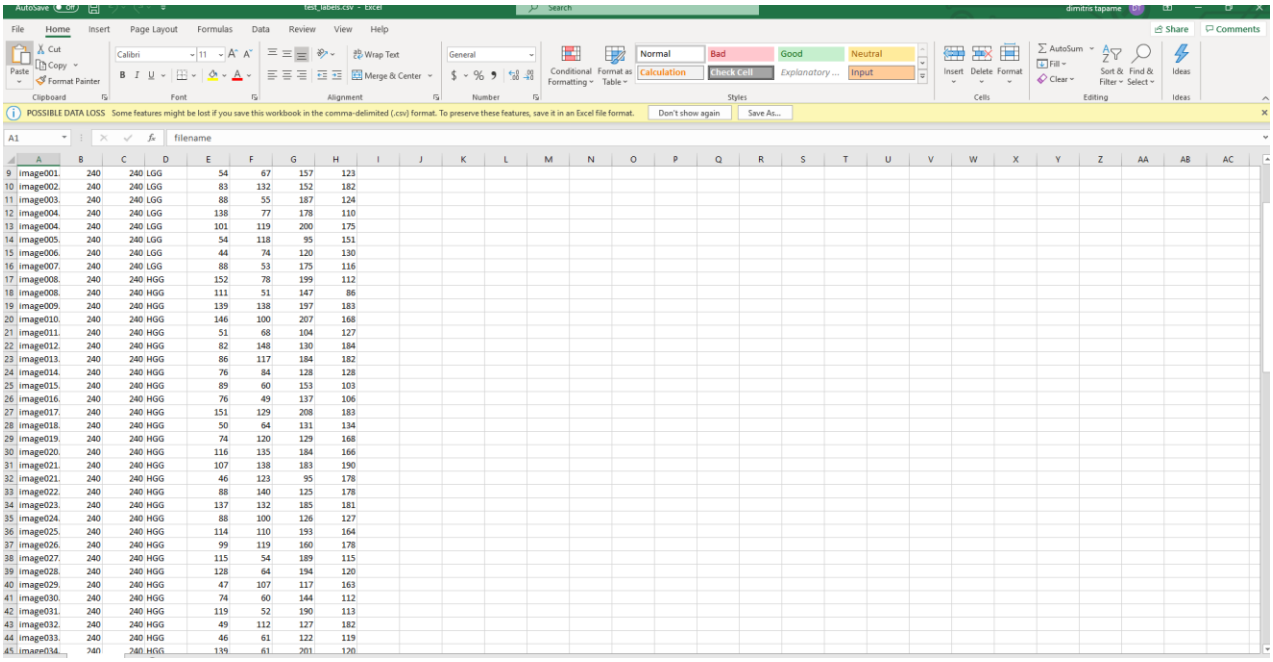
Τα ymin, ymax, xmin και xmax είναι οι συντεταγμένες του περιγράμματος του όγκου (LGG ή HGG, ανάλογα με το σύνολο που κάναμε το labelling) που κάναμε labelling.

6.6 Παραγωγή των δεδομένων εκπαίδευσης (TFRecords)

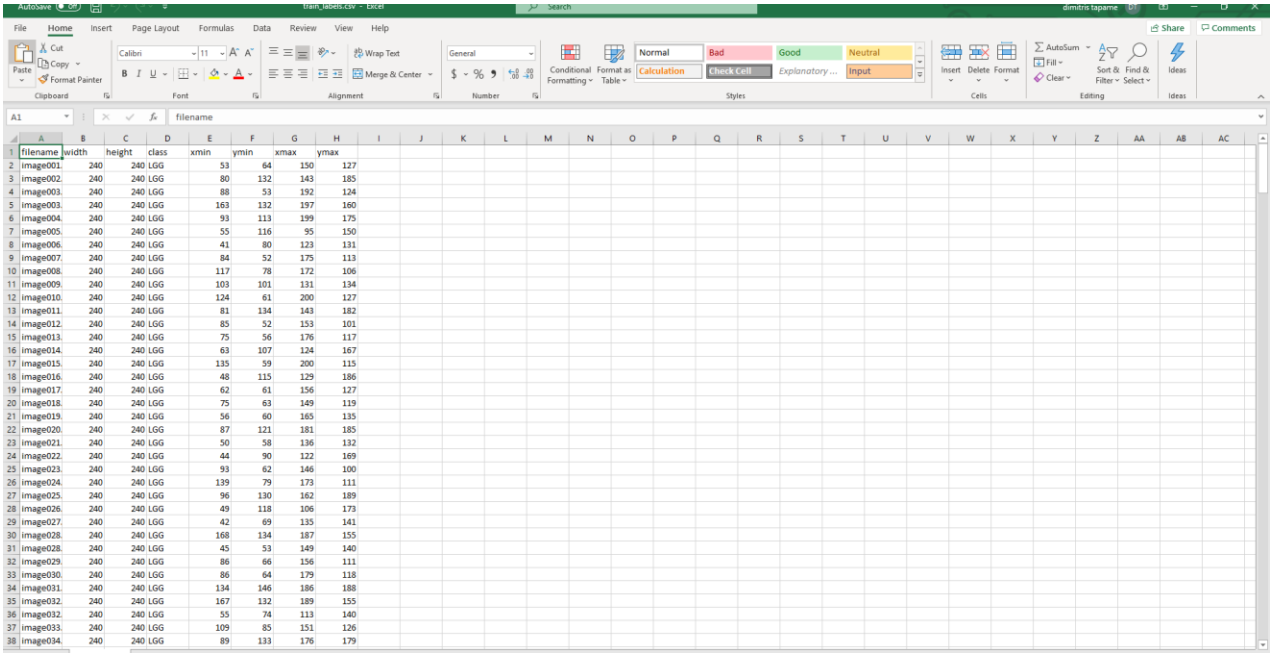
Μετά τις ετικέτες των εικόνων, δημιουργούμε τα TFRecords που χρησιμεύουν ως δεδομένα εισόδου στο μοντέλο εκπαίδευσης TensorFlow. Τα TFRecord είναι μια απλή μορφή για την αποθήκευση μιας ακολουθίας δυαδικών αρχείων. Τα protocol buffers είναι μια βιβλιοθήκη για αποτελεσματική σειριοποιήση δομημένων δεδομένων. Το tf.Example (ή protobuf) είναι τύπος μηνύματος που αντιπροσωπεύει το mapping (αντιστοίχιση) {string: value) κάθε αντικειμένου σε ένα αναγνωριστικό αριθμό. Πρώτον, τα δεδομένα .xml των εικόνων θα χρησιμοποιηθούν για τη δημιουργία αρχείων .csv που περιέχουν όλα τα δεδομένα για την εκπαίδευση και τις εικόνες δοκιμής, εκτελώντας το αρχείο xml_to_csv.py. Έτσι δημιουργήθηκαν τα αρχεία train_labels.csv και test_labels.csv στη θέση \object_detection\images.



Εικόνα 26: test_labels.csv και train_labels.csv



Εικόνα 27: test_labels.csv



filename	width	height	class	xmin	ymin	xmax	ymax
image001.	240	240	LGG	53	64	150	127
image002.	240	240	LGG	80	132	143	185
image003.	240	240	LGG	88	53	192	124
image003.	240	240	LGG	163	132	197	160
image004.	240	240	LGG	93	113	199	175
image005.	240	240	LGG	55	116	95	150
image006.	240	240	LGG	41	80	123	131
image007.	240	240	LGG	84	52	175	113
image008.	240	240	LGG	117	78	172	106
image009.	240	240	LGG	103	101	131	134
image010.	240	240	LGG	124	61	200	127
image011.	240	240	LGG	81	134	143	182
image012.	240	240	LGG	85	52	153	101
image013.	240	240	LGG	75	56	176	117
image014.	240	240	LGG	63	107	124	167
image015.	240	240	LGG	135	59	200	115
image016.	240	240	LGG	48	115	129	186
image017.	240	240	LGG	62	61	156	127
image018.	240	240	LGG	75	63	149	119
image019.	240	240	LGG	56	60	165	135
image020.	240	240	LGG	87	121	181	185
image021.	240	240	LGG	50	58	136	132
image022.	240	240	LGG	44	90	122	169
image023.	240	240	LGG	93	62	146	100
image024.	240	240	LGG	139	79	173	111
image025.	240	240	LGG	96	130	162	189
image026.	240	240	LGG	49	118	106	173
image027.	240	240	LGG	42	69	135	141
image028.	240	240	LGG	168	134	187	155
image028.	240	240	LGG	45	53	149	140
image029.	240	240	LGG	86	66	156	111
image030.	240	240	LGG	86	64	179	118
image031.	240	240	LGG	134	146	186	188
image032.	240	240	LGG	167	132	189	155
image032.	240	240	LGG	55	74	113	140
image033.	240	240	LGG	109	85	151	126
image034.	240	240	LGG	89	133	176	179

Εικόνα 28: train_labels.csv

Όπως αναφέραμε πριν τα csv περιέχουν όλες τις πληροφορίες των εικόνων του συνόλου δεδομένων, όπως το όνομα της jpg εικόνας, οι διαστάσεις της κάθε εικόνας, η κλάση του κάθε αντικειμένου προς αναγνώριση, καθώς και οι διαστάσεις του ορθογωνίου στο οποίο ανήκει το αντικείμενο προς ανίχνευση.

Στη συνέχεια, στο αρχείο generate_tfrecord.py βάζουμε τον αριθμό ετικετών, όπου σε κάθε αντικείμενο έχει εκχωρηθεί ένας αναγνωριστικός αριθμός.

```
def class_text_to_int(row_label):
    if row_label == 'LGG':
        return 1
    elif row_label == 'HGG':
        return 2
    else:
        None
```

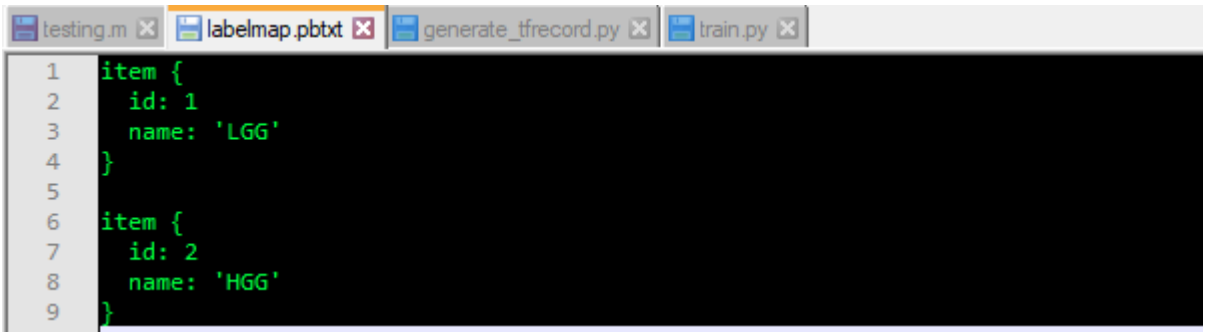
Εικόνα 29: label map

Τέλος παράγουμε τα TFRecord (train.record και test.record) από το φάκελο \research\object_detection χρησιμοποιώντας τις ακόλουθες εντολές:

```
python generate_tfrecord.py --csv_input=images\train_labels.csv --image_dir=images\train --output_path=train.record
python generate_tfrecord.py --csv_input=images\test_labels.csv --image_dir=images\test --output_path=test.record
```

6.7 Δημιουργία αντιστοίχισης ετικετών και διαμόρφωση εκπαίδευσης

Η αντιστοίχιση ετικετών λέει στον εκπαιδευτή τι είναι κάθε αντικείμενο καθορίζοντας μια αντιστοίχιση των ονομάτων της κάθε κατηγορίας σε αριθμούς αναγνώρισης τάξης. Στο notepad++ δημιουργούμε ένα αρχείο labelmap.pbtxt (μορφή κειμένου αντί .pb που είναι binary format) στον φάκελο C: \ tensorflow1 \ models \ research \ object_detection \ training



Εικόνα 30: labelmap.pbtxt

Οι αριθμοί αναγνωριστικών του labelmap πρέπει να είναι ίδιοι με αυτούς που ορίζονται στο αρχείο generate_tfrecord.py

Τέλος, πρέπει να διαμορφωθεί το object detection pipeline, όπου καθορίζει το μοντέλο και τις παραμέτρους που θα χρησιμοποιηθούν για την εκπαίδευση. Αυτό είναι το τελευταίο βήμα πριν από την εκπαίδευση.

Στο C: \ tensorflow1 \ models \ research \ object_detection \ samples \ configs και αντιγράψαμε το αρχείο faster_rcnn_inception_v2_pets.config στον κατάλογο \ object_detection \ training. Στη συνέχεια, ανοίγουμε το αρχείο με ένα πρόγραμμα επεξεργασίας κειμένου (notepad++). Υπάρχουν αρκετές αλλαγές στο αρχείο .config, αλλάζουμε τον αριθμό των κλάσεων και των παραδειγμάτων και προσθέτοντας τις διαδρομές των αρχείων στα δεδομένα εκπαίδευσης.

- Αλλαγή της κλάσης num_classes στον αριθμό των διαφορετικών αντικειμένων που θέλουμε να ανιχνεύσει ο ταξινομητής.

```
model {
  faster_rcnn {
    num_classes: 2
    image_resizer {
      keep_aspect_ratio_resizer {
        min_dimension: 600
        max_dimension: 1024
      }
    }
    feature_extractor {
```

- Στην ενότητα train_input_reader, αλλάζουμε το input_path και το path_map_path to:

  input_path: "C: /tensorflow1/models/research/object_detection/train.record"

  path_map_path:"C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"

```
train_input_reader: {
  tf_record_input_reader {
    input_path: "C:/tensorflow1/models/research/object_detection/train.record"
  }
  label_map_path: "C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"
}
```

- Αλλαγή num_examples στον αριθμό των εικόνων \ tests \

```
eval_config: {
  metrics_set: "coco_detection_metrics"
  num_examples: 86
}
```

- Στην ενότητα eval_input_reader, αλλάζουμε τη διαδρομή εισόδου και το path_map_path

  input_path: "C: /tensorflow1/models/research/object_detection/test.record"

  path_map_path:"C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"

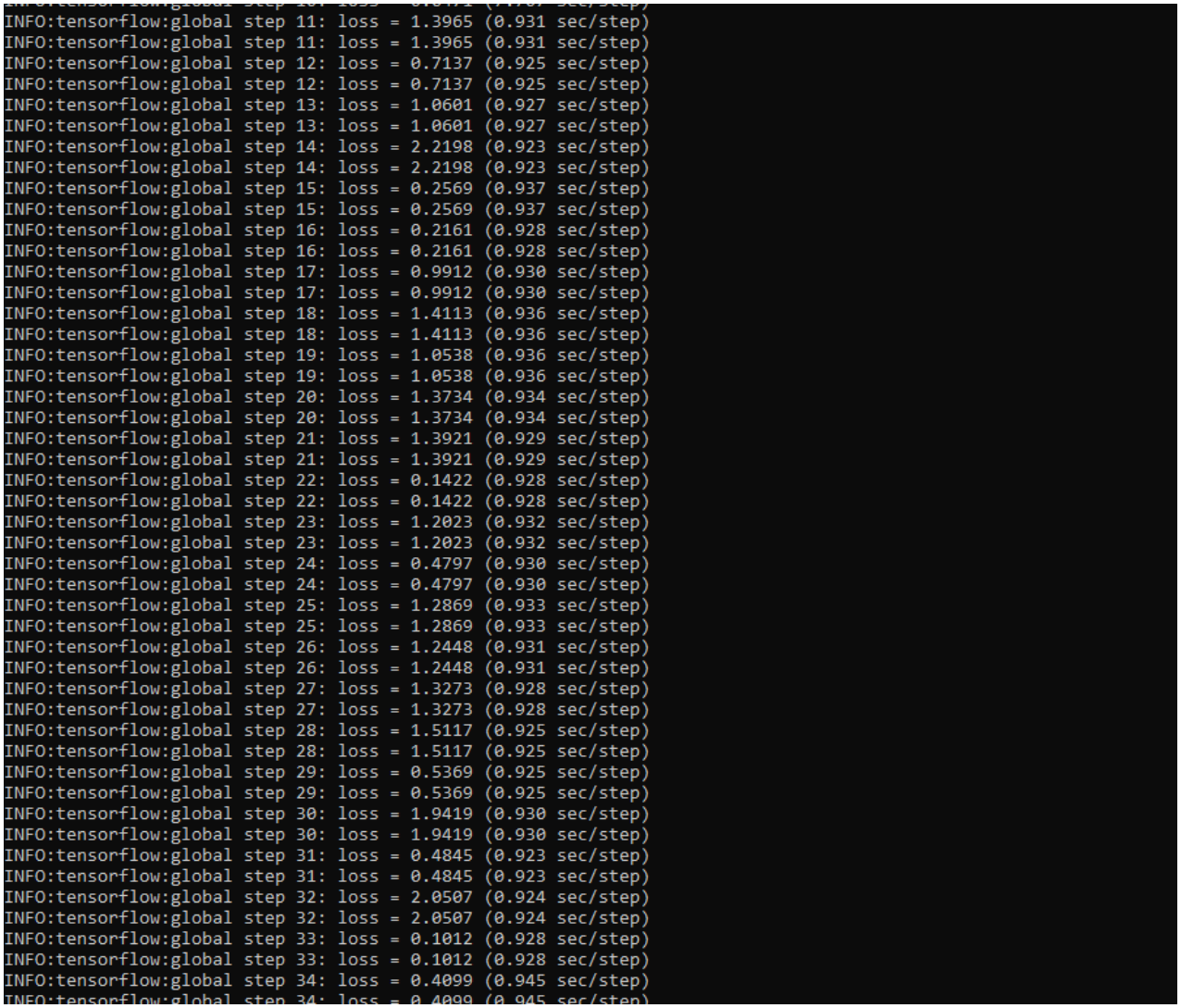
```
eval_input_reader: {
  tf_record_input_reader {
    input_path: "C:/tensorflow1/models/research/object_detection/test.record"
  }
  label_map_path: "C:/tensorflow1/models/research/object_detection/training/labelmap.pbtxt"
```

6.8 Εκπαίδευση του μοντέλου μας

Εκτελούμε το script train.py

python train.py --logtostderr --train_dir=training/ --
pipeline_config_path=training/faster_rcnn_inception_v2_pets.config

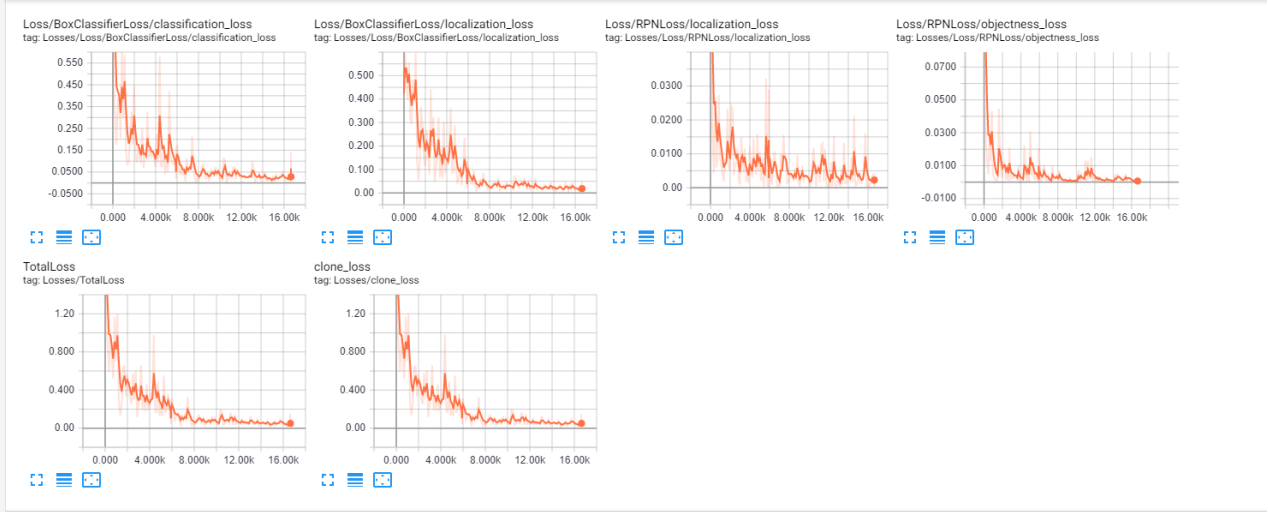
το TensorFlow θα προετοιμάσει την αρχικοποίηση της εκπαίδευσης πριν ξεκινήσει η πραγματική εκπαίδευση, όπως φαίνεται στο παρακάτω στιγμιότυπα.



Εικόνα 31: Εκπαίδευση του δικτύου

Κάθε βήμα της εκπαίδευσης αναφέρει την απώλεια. Θα ξεκινήσει ψηλά και θα μειωθεί όσο προχωράει η εκπαίδευση. Για την εκπαίδευσή στο μοντέλο Faster-RCNN-Inception-V2, αφήσαμε το μοντέλο να εκπαιδευτεί έως ότου η απώλεια υποχωρήσει σταθερά κάτω από 0,05 (βλ. παρακάτω), η οποία διήρκησε περίπου 5,5 ώρες. Μπορούμε να δούμε την πρόοδο της εκπαίδευσης χρησιμοποιώντας το TensorBoard. Για να το κάνουμε αυτό, ανοίγουμε νέο παράθυρο του Anaconda Prompt, ενεργοποιούμε το εικονικό περιβάλλον tensorflow1, και στον κατάλογο C: \ tensorflow1 \ models \ research \ object_detection εκτελούμε την εντολή :

```
(tensorflow1)
C:\tensorflow1\models\research\object_detection>tensorboard --
logdir=training
```



Εικόνα 32: Tensorboard losses

Στο step 17,3k το Loss/BoxClassifier/classification__loss είναι 0.07806, το BoxClassifier/localization __loss είναι 0.05369, το RPNLoss / localization __loss είναι $6.8439e^{-3}$, το RPNLoss/ objectness __loss είναι $1.1506e^{-3}$. Το loss είναι το άθροισμα των classification__loss και localization __loss. Το loss δηλαδή αποτελείται από δύο μέρη: την απώλεια εντοπισμού για την πρόβλεψη αντιστάθμισης του κιβώτιου οριοθέτησης και την απώλεια ταξινόμησης για πιθανότητες κλάσης υπό όρους. Και τα δύο υπολογίζονται ως άθροισμα τετραγωνικών σφαλμάτων. Το global step είναι η επανάληψη (πχ global step 10 σημαίνει ότι επεξεργαζόμαστε την 10 παρτίδα (batch), global step/sec είναι το ίδιο πράγμα με το sec/step. Το RPN loss είναι ένας συνδυασμός του classification και του regression loss, που έχουν αναφερθεί στο κεφάλαιο 4.1 στην παράγραφο για το faster R-CNN.

Η εκπαίδευση περιοδικά αποθηκεύει τα σημεία ελέγχου κάθε πέντε λεπτά. Το σημείο ελέγχου στον μεγαλύτερο αριθμό βημάτων θα χρησιμοποιηθεί για τη δημιουργία του παγωμένου γραφήματος συμπερασμάτων (frozen inference graph).

6.9 *Μετρήσεις αξιολόγησης (evaluation metrics)*

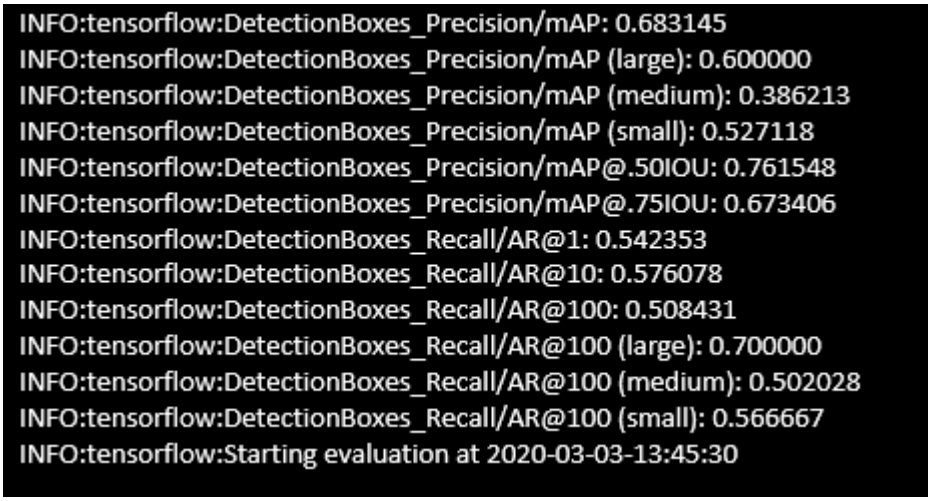
Για να αξιολογήσουμε το μοντέλο μας τρέχουμε την εντολή eval.py από τον φάκελο object_detection. Η αξιολόγηση θα πραγματοποιήσει προβλέψεις χρησιμοποιώντας τις διαθέσιμες εικόνες σε έναν συγκεκριμένο κατάλογο στο σύνολο δοκιμών (test). Υπάρχουν δύο διαφορετικές αποστολές μέτρησης: προσδιορισμός του αν υπάρχει αντικείμενο στην εικόνα (ταξινόμηση-

classification), καθορισμός της θέσης του αντικειμένου (εντοπισμός-localization, εργασία παλινδρόμησης).

python eval_util_test.py --logtostder --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config –checkpoint_dir=training/ --eval_dir=eval/

```
INFO:tensorflow:Running eval batches done.
INFO:tensorflow:# success: 86
INFO:tensorflow:# skipped: 0
INFO:tensorflow:Performing evaluation on 86 images.
creating index...
index created!
INFO:tensorflow:Loading and preparing annotation results...
INFO:tensorflow:DONE (t=0.02s)
creating index...
index created!
Running per image evaluation...
Evaluate annotation type *bbox*
DONE (t=0.71s).
Accumulating evaluation results...
DONE (t=0.16s).
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.683
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.762
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.673
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.527
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.386
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.600
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.542
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.676
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.608
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.567
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.502
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.700
INFO:tensorflow:Writing metrics to tf summary.
```

Εικόνα 33: Evaluation metrics (recall and precision)



Εικόνα 34: Evaluation metrics (recall and precision)1

Methods	*AP*	*AP^{50}*	*AP^{75}*	*AP_{small}*	*AP_{medium}*	*AP_{large}*
Faster R-CNN	**0.683**	**0.762**	**0.673**	**0.527**	**0.386**	**0.600**

Methods	*$AR_{max=1}$*	*$AR_{max=10}$*	*$AR_{max=100}$*	*AR_{small}*	*AR_{medium}*	*AR_{large}*
Faster R-CNN	**0.542**	**0.676**	**0.608**	**0.567**	**0.502**	**0.700**

```
Average Precision (AP):
  AP                % AP at IoU=.50:.05:.95 (primary challenge metric)
  AP^IoU=.50        % AP at IoU=.50 (PASCAL VOC metric)
  AP^IoU=.75        % AP at IoU=.75 (strict metric)
AP Across Scales:
  AP^small          % AP for small objects: area < 32^2
  AP^medium         % AP for medium objects: 32^2 < area < 96^2
  AP^large          % AP for large objects: area > 96^2
Average Recall (AR):
  AR^max=1          % AR given 1 detection per image
  AR^max=10         % AR given 10 detections per image
  AR^max=100        % AR given 100 detections per image
AR Across Scales:
  AR^small          % AR for small objects: area < 32^2
  AR^medium         % AR for medium objects: 32^2 < area < 96^2
  AR^large          % AR for large objects: area > 96^2
```

Υπολογισμός Precision και Recall

Το AP είναι ο μέσος όρος για όλες τις κατηγορίες. Παραδοσιακά, αυτό ονομάζεται "μέση ακρίβεια" (mAP), το οποίο πάντοτε υπολογίζεται σε ένα σύνολο αντικειμένων. η βαθμολογία AP ορίζεται ως η μέση ακρίβεια στο σύνολο των 11 ισομερώς recall τιμών.

Με απλά λόγια, η υψηλή ακρίβεια (precision) σημαίνει ότι ένας αλγόριθμος επιστρέφει το ποσοστό των θετικών προβλέψεων που είναι σωστό , ενώ η υψηλή ανάκληση (recall) σημαίνει ότι ένας αλγόριθμος επέστρεψε τα περισσότερα από τα σχετικά αποτελέσματα. Όπου TP=true positive, FP=false positive, FN=false negative.

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

Το maxDet είναι το max detection ανά εικόνα παίρνοντας τις ακόλουθες τιμές 1,10 ή 100 ανά εικόνα, Δηλαδή η μέγιστη ανάκληση με 1 ανίχνευση ανά πλαίσιο , 10 ή 100 αντίστοιχα. Όλες οι μετρικές υπολογίζονται επιτρέποντας το πολύ 100 ανίχνευσης κορυφαίας βαθμολογίας ανά εικόνα (σε όλες τις κατηγορίες). Το max detection αντικειμένων σε κάθε εικόνα με το label που έχουμε κάνει δεν ξεπερνάει το 2 γι' αυτό έχουμε το ίδιο αποτέλεσμα στο maxdet=10 και στο maxdet=100.

IoU (Intersection over union) μετρά πόση αλληλοεπικάλυψη υπάρχει μεταξύ των 2 περιοχών (ανιχνεύσιμου πλαισίου και του ground truth), αυτό μετρά πόσο καλή είναι η πρόβλεψή μας στον ανιχνευτή αντικειμένων με το ground truth (το πραγματικό όριο αντικειμένου), δηλαδή IoU=τομή των δύο περιοχών/ένωση περιοχών. Για να θεωρηθεί μια σωστή ανίχνευση, η περιοχή επικάλυψης μεταξύ του ανιχνευόμενου πλαισίου οριοθέτησης και του πλαισίου οριοθέτησης του ground truth πρέπει να υπερβαίνει το 50% που δίνεται από τον τύπο που μόλις αναφέραμε. Η μεταβολή των επιπέδων εμπιστοσύνης παράγει διαφορετικές τιμές ακρίβειας και ανάκλησης. Αν αυξήσετε το κατώφλι εμπιστοσύνης, ο αριθμός των πλαισίων οριοθέτησης που εντοπίζονται θα είναι μικρότερος αλλά ακριβέστερος.

Καθώς αυξάνουμε το "κατώτατο όριο-threshold", θα αυξήσουμε γενικά το ποσοστό των αληθινών θετικών (θα συμπεριληφθούν πιο πιθανά πλαίσια οριοθέτησης) σε βάρος της συμπερίληψης πολλών περισσότερων ψευδών θετικών. Ταυτόχρονα, ο αριθμός των ψευδών αρνητικών θα μειωθεί δραστικά. Η ακρίβεια θα μειωθεί, ενώ η ανάκληση αυξάνεται. Η τιμή ανάκλησης αυξάνεται καθώς συμπεριλαμβάνουμε περισσότερες προβλέψεις, αλλά η ακρίβεια θα διατηρηθεί αρχικά και θα μειωθεί στη συνέχεια. Είναι πολύ σημαντικό να σημειώσουμε ότι υπάρχει μια αντίστροφη σχέση μεταξύ της ακρίβειας και της ανάκλησης και ότι αυτές οι μετρήσεις εξαρτώνται από το όριο βαθμολογίας μοντέλου (threshold) που ορίσαμε (καθώς και φυσικά από την ποιότητα του μοντέλου).

Το PASCAL VOC είναι ένα δημοφιλές σύνολο δεδομένων για την ανίχνευση αντικειμένων. Για την πρόκληση PASCAL VOC, μια πρόβλεψη είναι θετική αν IoU> 0,5 (1 IoU). Ωστόσο, αν εντοπιστούν πολλαπλές ανιχνεύσεις του ίδιου αντικειμένου, μετράει το πρώτο ως θετικό ενώ το υπόλοιπο ως αρνητικό.

Για το COCO, το AP είναι ο μέσος όρος των πολλαπλών IoU. **Το AP @ [.5: .95]** αντιστοιχεί στο μέσο AP για το IoU από 0,5 έως 0,95 με μέγεθος βήματος 0,05, το AP είναι ο μέσος όρος πάνω από 10 IoU επίπεδα σε 80 κατηγορίες.

6.10 Εξαγωγή γραφήματος συμπερασμάτων (inference graph)

Τώρα που η εκπαίδευση είναι πλήρης, το τελευταίο βήμα είναι να δημιουργηθεί το τελικό γράφημα συμπερασμάτων (frozen inference graph .pb file) μέσα από το φάκελο \object_detection εκτελούμε την ακόλουθη όπου το model.ckpt-17158 έχει τον υψηλότερο αριθμό εκπαίδευσης στο φάκελο training με την εντολή:

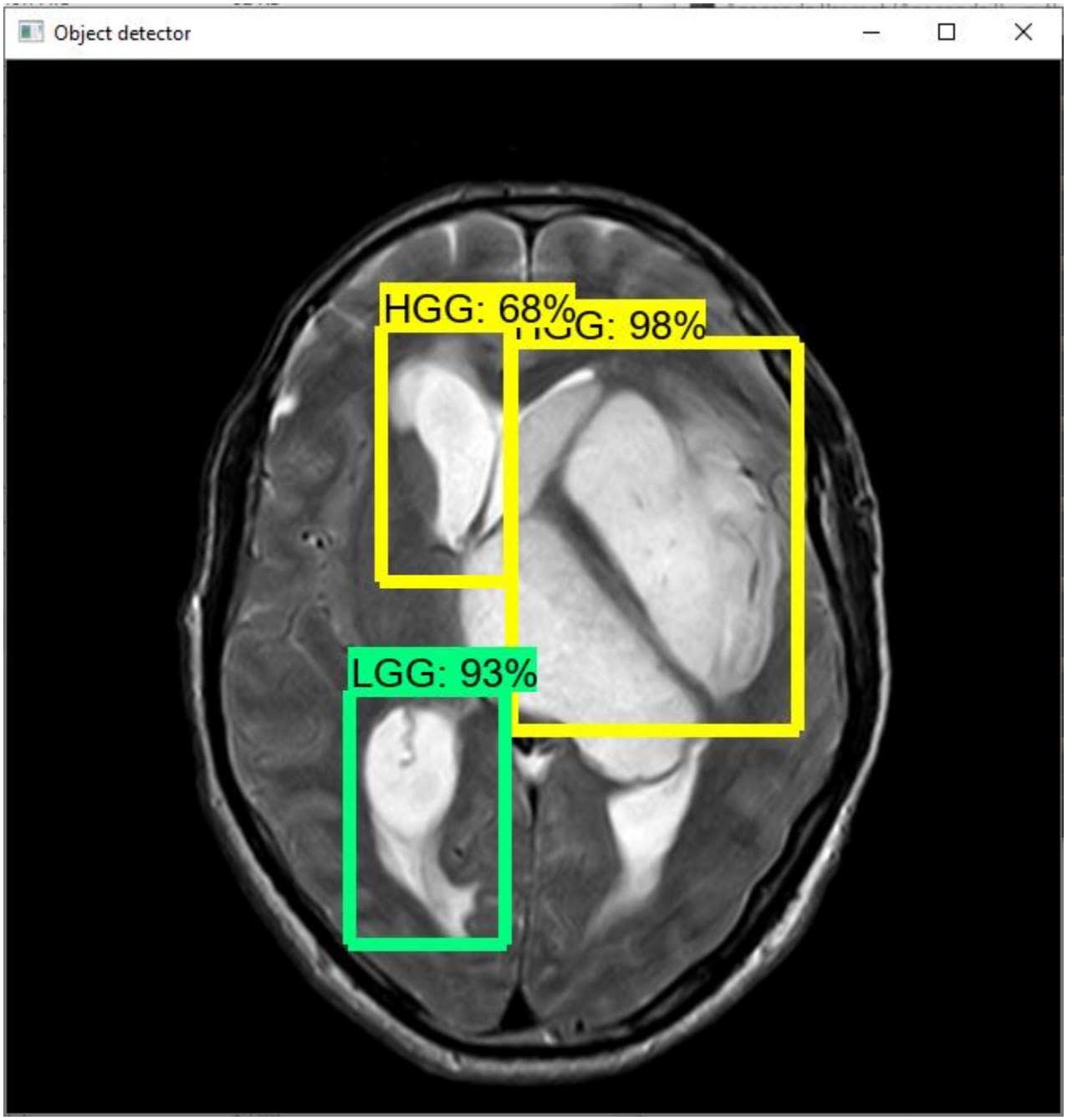
```
python export_inference_graph.py --input_type image_tensor --pipeline_config_path
training/faster_rcnn_inception_v2_pets.config--trained_checkpoint_prefix
training/model.ckpt-120670 --output_directory inference_graph
```

Αυτό δημιουργεί ένα αρχείο frozen_inference_graph.pb στο φάκελο \ object_detection \ inference_graph. Το αρχείο .pb περιέχει τον ταξινομητή ανίχνευσης αντικειμένων.

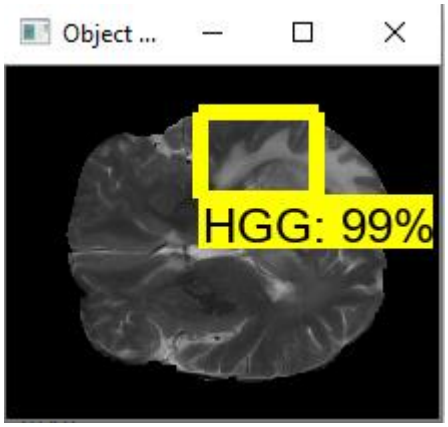
6.11 Έλεγχος του ταξινομητή ανίχνευσης αντικειμένων που δημιουργήθηκε

Ο ταξινομητής ανίχνευσης αντικειμένων είναι έτοιμος να ξεκινήσει. Έχουμε scripts Python για να το ελέγξουμε σε μια εικόνα, βίντεο ή web κάμερα. Για να ελέγξουμε τον ανιχνευτή αντικειμένων, μετακινούμε μια εικόνα του αντικειμένου ή των αντικειμένων στο φάκελο \ object_detection και αλλάζουμε τη μεταβλητή IMAGE_NAME στο Object_detection_image .py για να ταιριάζει με το όνομα του αρχείου της εικόνας. Εναλλακτικά, μπορούμε να χρησιμοποιήσουμε ένα βίντεο των αντικειμένων (χρησιμοποιώντας Object_detection_video.py) ή απλώς μια κάμερα web και (χρησιμοποιώντας Object_detection_webcam.py).Ο ανιχνευτής αντικειμένου θα προετοιμαστεί για περίπου 10-15 δευτερόλεπτα και στη συνέχεια θα εμφανιστεί ένα παράθυρο το οποίο θα δείχνει που υπάρχει όγκος και εάν είναι LGG ή HGG στην εικόνα. Οι εικόνες που χρησιμοποιήσαμε για την πρόβλεψη του μοντέλου είναι από το dataset του Brats2013, καθώς δεν υπήρχαν πολλές MRI εικόνες(png ή jpg) στο διαδίκτυο. Στο συγκεκριμένο dataset οι εικόνες ήταν διαχωρισμένες σε LGG και HGG, πράγμα το οποίο μας διευκόλυνε για την αξιολόγηση των αποτελεσμάτων μας.

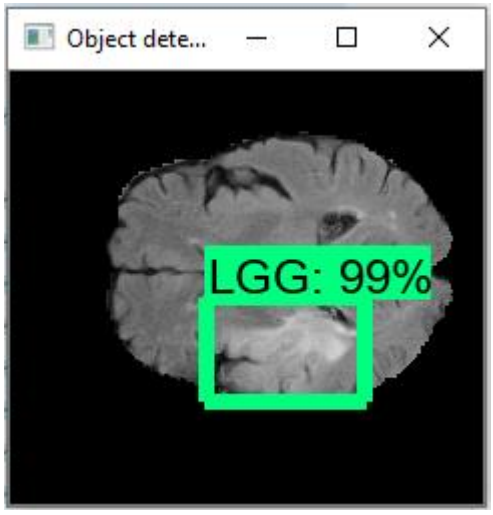
Στην εικόνα 35 παρατηρούμε ότι η αναγνώριση του μοντέλου μας για το που υπάρχει όγκος ανεξάρτητα από το είδος του (LGG-HGG), γίνεται με επιτυχία. Στις εικόνες 36 και 37 βλέπουμε την επιτυχημένη αναγνώριση όγκου αλλά και το είδος του. Τέλος, στις εικόνες 38 και 39 βλέπουμε επιτυχημένη αναγνώριση όγκου, αλλά την ανεπιτυχή αναγνώριση για το είδος του.



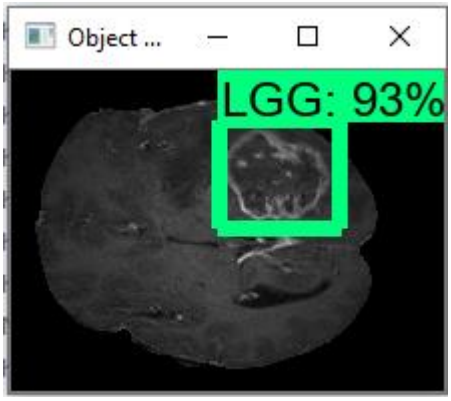
Εικόνα 35: Πολλαπλή αναγνώριση όγκων στον εγκέφαλο



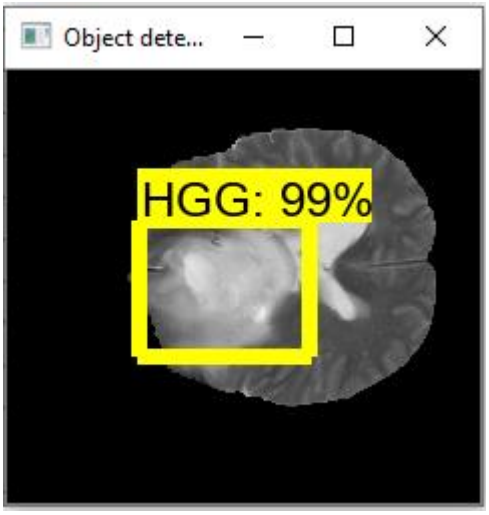
Εικόνα 36:Επιτυχής αναγνώριση HGG



Εικόνα 37: Επιτυχής αναγνώριση LGG

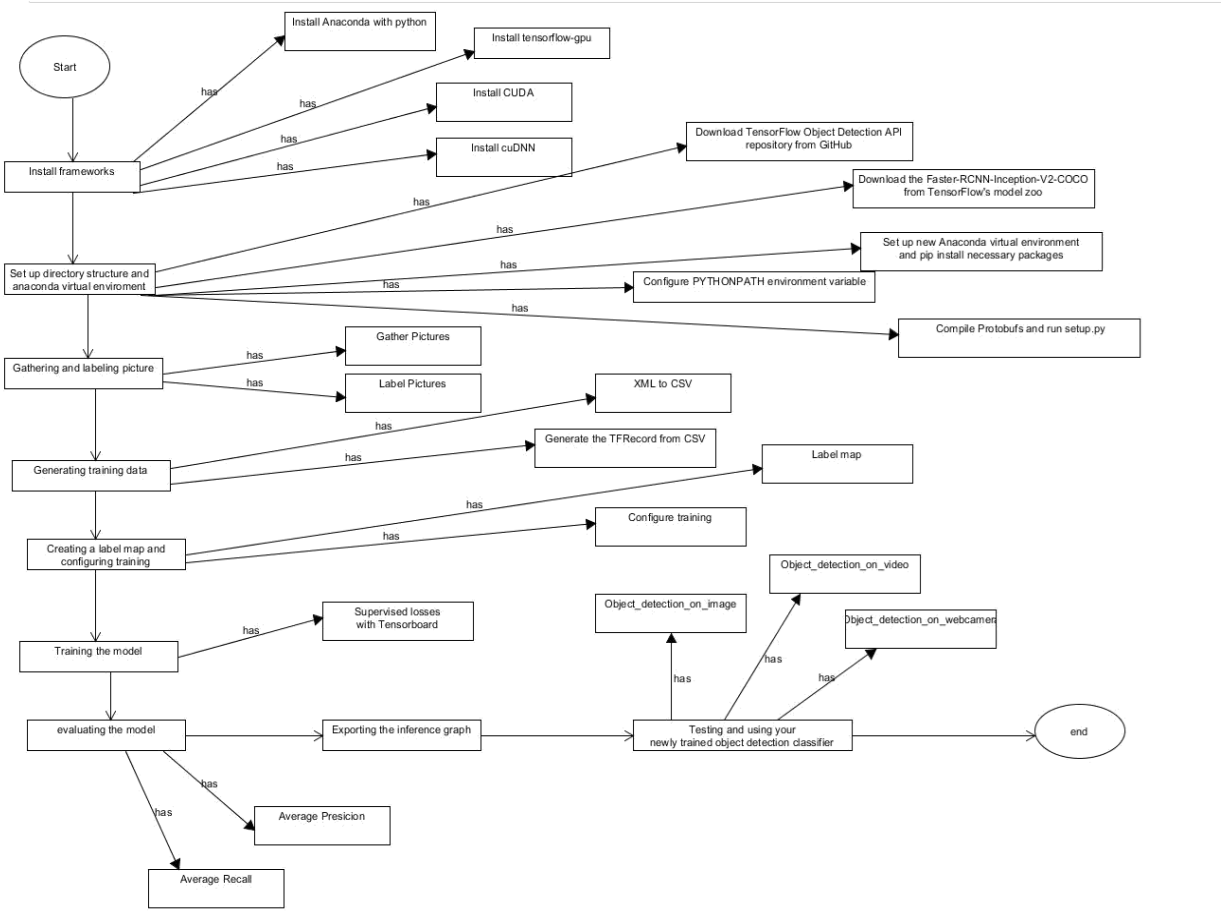


Εικόνα 38: Ανεπιτυχής αναγνώριση είδους του όγκου



Εικόνα 39:Ανεπιτυχής αναγνώριση είδος του όγκου(1)

Καθώς σκανάρει την εικόνα, εάν υπάρχει όγκος θα μαρκάρει το μέρος όπου είναι ο όγκος με ένα τετράγωνο και θα εμφανίζει ένα label LGG(%) ή HGG(%). Το αποτέλεσμα είναι αρκετά ικανοποιητικό δεδομένου ότι το δίκτυο έχει εκπαιδευτεί μόνο με 470 εικόνες για LGG και HGG .



Εικόνα 40: Διάγραμμα της υλοποίησης

7 *Συμπεράσματα*

Στην παρούσα εργασία περιγράψαμε βασικές αρχές της αναγνώρισης αντικειμένου που σχετίζονται διάφορους αλγόριθμους οπτικής ανίχνευσης αντικειμένων (όπως Template Matching, β) το Support Vector Machine και γ) τα Artificial Neural Network Algorithms) συμπεριλαμβανομένων μερικών από τους state-of-the-art ανιχνευτών CNN που θα μπορούσαν να χρησιμοποιηθούν για ανίχνευση όγκου στον εγκέφαλο σε μη-πραγματικό χρόνο. Από αυτούς τους ανιχνευτές, χρησιμοποιήσαμε το faster r-cnn σύστημα, το οποίο αποδείχθηκε αποτελεσματικό και εύκολο να τροποποιηθεί για εφαρμογή με χρήση tensorflow gpu, python, cuda και cudnn. Έχουμε εξηγήσει τις τροποποιήσεις που έγιναν στο σύστημα faster r-cnn. Ο ανιχνευτής λειτουργεί καλά σε μαγνητικές τομογραφίες με την μορφή png ή jpg. Μετά την εκπαίδευση δεν προλάβαμε να αξιολογήσουμε το μοντέλο μας. Αυτό το οποίο θα κάναμε ήταν να υπολογίσουμε μετρικές αξιολόγησης recall και precision, όπου η υψηλή ακρίβεια (precision) σημαίνει ότι ένας αλγόριθμος επέστρεψε σημαντικά πιο σχετικά αποτελέσματα από ότι άσχετα, ενώ η υψηλή ανάκληση (recall) σημαίνει ότι ένας αλγόριθμος επέστρεψε τα περισσότερα από τα σχετικά αποτελέσματα.

Τέλος ελέγξαμε τον ταξινομητή ανίχνευσης αντικειμένων που δημιουργήθηκε σε εικόνες που βρήκαμε από τον διαγωνισμό Brats2013, το οποίο διαχωρίζει τον όγκο από LGG σε HGG, ώστε να είναι έγκυρα τα αποτελέσματα που θα πάρουμε. Καθώς σκαν-άρει την εικόνα εμφανίζει ένα label LGG(%) ή HGG(%) ,όπου υπάρχει όγκος . Το αποτέλεσμα είναι ικανοποιητικό δεδομένου ότι το δίκτυο έχει εκπαιδευτεί μόνο με 470 εικόνες για LGG και HGG με αρκετά καλή ακρίβεια εμφάνισης (που υπάρχει ο όγκος).

- Το μοντέλο που δημιουργήσαμε ανιχνεύει σχεδόν πάντα που υπάρχει όγκος στον εγκέφαλο.
- Το μοντέλο που δημιουργήσαμε δεν κάνει πάντα σωστή αναγνώριση του είδους του όγκου. Υπάρχουν 2 πολύ σημαντικοί παράγοντες για την λανθασμένη αναγνώριση του είδους.
    - Ο ένας είναι λόγο του πολύ μικρού dataset που χρησιμοποιήσαμε για την εκπαίδευση του δικτύου ,καθώς μετά από την μετατροπή που κάναμε στα αρχεία .mha για να παράγουμε .jpg εικόνες, υπήρχαν πολλές αλλοιώσεις σε πολύ μεγάλο μέρος του dataset. Έτσι αναγκαστήκαμε να ξεχωρίσουμε χειροκίνητα τις εικόνες όπου εμφανίζεται ξεκάθαρα ο όγκος (LGG-HGG) για να μην είναι λανθασμένα τα labelling και μπερδευτεί το μοντέλο μας.
    - Όπως προαναφέραμε και στο κεφάλαιο 6.4 είναι πολύ δύσκολο να γίνει διάκριση σε κάθε περίπτωση LGG και HGG, με βάση μόνο τα φαινότυπα που είναι ορατά στο ανθρώπινο μάτι. Για τον λόγο αυτό, τα χαρακτηριστικά που αντλούνται από το

δίκτυο πρότασης περιοχής (RPN) ενός Faster-RCNN θα ήταν χρήσιμα στην διαφοροποίηση μεταξύ τους. Με την μετατροπή που κάναμε δεν είχαμε μόνο απώλειες εικόνων από το dataset, αλλά είχαμε και αλλοιώσεις στα pixels των εικόνων, κάτι το οποίο παίζει καθοριστικό παράγοντα για το νευρωνικό μας ,ειδικά στη συγκεκριμένη υλοποίηση (αναγνώρισης όγκου στον εγκέφαλο), όπου η διαφοροποίηση μεταξύ τους βασίζεται στα pixels .

- Η δημιουργία του ανιχνευτή αντικείμενου και η ανίχνευση αντικείμενων είναι διαδικασίες που απαιτούν πολλές υπολογιστικές πράξεις. Αυτό σημαίνει για να αποφευχθεί η υπερχείλιση της μνήμης και να γίνουν πιο γρήγορα οι υπολογισμοί πρέπει να διαθέτουμε καλή κάρτα γραφικών.

Σε μελλοντική βελτίωση της συγκεκριμένης εργασίας θα μπορούσαμε να ορίζαμε μεγαλύτερη ποικιλία εικόνων, καθώς και μια δημιουργία μεγαλύτερου συνόλου δεδομένων. Αυτό θα είχε σαν αποτέλεσμα μια μεγαλύτερη αξιοπιστία στην εξαγωγή συμπερασμάτων του μοντέλου μας.

Βιβλιογραφία

Ren, F.X., Huang, J.S., Jiang, R.Y., Klette, R.: General Traffic Sign Recognition by Feature Matching. In IVCNZ, σσ.409--414(2009)

Li, Z.F., Tang, X.O.: Bayesian Face Recognition Using Support Vector Machine and Face Clustering. In CVPR, vol. 2, σσ.374—380, Washington, DC (2004)

Torres, M., Hervás, C., García, C.: Multinomial Logistic Regression and Product Unit Neural Network Models: Application of a New Hybrid Methodology for Solving a Classification Problem in The Livestock Sector. Expert Systems with Applications, 36, σσ.12225--12235(2009)

Banerjee, Subhashis, et al. "Deep Radiomics for Brain Tumor Detection and Classification from Multi-Sequence MRI." arXiv preprint arXiv:1903.09240 (2019).

Girshick, R., Donahue, J., Darrell, T., and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (2014), σσ. 580-587.

Walther, D., Itti, L., Riesenhuber, M., Poggio, T., andKoch, C. Attentional selection for object recognition - a gentle way. In International Workshop on Biologically Motivated Computer Vision (2002), Springer, σσ. 472-479.

Y. Wu, J. Lim, and M.-H. Yang, “Online object tracking: A benchmark,” in Proceed-ings of the IEEE conference on computer vision and pattern recognition, σσ. 2411– 2418, 2013.

R. A. Khushboo Khurana, “Techniques for object recognition in images and multi-object detection,” International Journal of Advanced Research in Computer Engineering & Technology, vol. 2, apr 2013.

T. Gevers and A. W. Smeulders, “Color-based object recognition,” Pattern Recognition, vol. 32, no. 3, σσ. 453 – 464, 1999.

Z. Kalal, K. Mikolajczyk, and J. Matas, “Tracking-learning-detection,” IEEE trans-actions on pattern analysis and machine intelligence, vol. 34, no. 7, σσ. 1409– 1422, 2012.

G. Nebehay and R. Pflugfelder, “Clustering of Static-Adaptive correspondences for deformable object tracking,” in Computer Vision and Pattern Recognition, σσ. 2784– 2791, IEEE, June 2015

J. W. Howarth, H. H. C. Bakker, and R. C. Flemmer, “Feature-based object recognition,” in 2009 4th International Conference on Autonomous Robots and Agents, σσ. 375–379, Feb 2009.

H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in European conference on computer vision, σσ. 404–417, Springer, 2006.

J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

B. K. Horn and B. G. Schunck, “Determining optical flow,” Artificial intelligence, vol. 17, no. 1-3, σσ. 185–203, 1981.

Pereira, Sérgio, et al. "Brain tumor segmentation using convolutional neural networks in MRI images." IEEE transactions on medical imaging 35.5 (2016): 1240-1251.

Seetha, J., and S. Selvakumar Raja. "Brain tumor classification using convolutional neural networks." Biomedical & Pharmacology Journal 11.3 (2018): 1457.

S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in Computer Vision (ICCV), 2011 IEEE International Conference on, σσ. 2548–2555, IEEE, 2011

Russell, Stuart, Norvig, Peter. Artificial Intelligence: A Modern Approach. 3rd edition. Prentice Hall, 2009. ISBN 0-13-604259-7.

Szeliski, Richard. Computer Vision: Algorithms and Applications [online]. Springer-Verlag, 2010. ISBN 978-1848829343 [13.7.2014]. Available from: http://www.szeliski.org/Book/

A. Krizhevsky, L. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Proc. Neural Information Processing Systems, 2012

C. Stergiou and D. Siganos, “Neural networks.” Available at:
http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html.

C. M. Bishop, Neural networks for pattern recognition. Oxford university press, 11 1995.

M. Y. W. Teow, “Understanding convolutional neural networks using aminimal model for handwritten digit recognition,” in 2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS), σσ. 167–172, Oct 2017.

Y. LeCun, K. Kavukcuoglu, and C. Farabet, “Convolutional networks and applications in vision,” in Proceedings of 2010 IEEE International Symposium on Circuits and Systems, σσ. 253–256, May 2010.

Jasper RR Uijlings ``Selective search for object recognition". Στο: International journal of computer vision 104.2 (2013), σσ. 154–171.

S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, σσ. 91–99, 2015.

R. Girshick, “Fast r-cnn,” arXiv preprint arXiv:1504.08083, 2015.

M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International journal of computer vision, vol. 88, no. 2, σσ. 303–338, 2010.

W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in European conference on computer vision, σσ. 21–37, Springer, 2016.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), σσ. 779–788, June 2016.

Afshar, Parnian, Arash Mohammadi, and Konstantinos N. Plataniotis. "Brain tumor type classification via capsule networks." 2018 25th IEEE International Conference on Image Processing (ICIP). IEEE, 2018.

J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), σσ. 6517– 6525, July 2017.

Abadi, M.; Agarwal, A.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015.

Louis, David N., et al. "The 2016 World Health Organization classification of tumors of the central nervous system: a summary." Acta neuropathologica 131.6 (2016): 803-820.

Van den Bent, Martin J., et al. "Adjuvant procarbazine, lomustine, and vincristine chemotherapy in newly diagnosed anaplastic oligodendroglioma: long-term follow-up of EORTC brain tumor group study 26951." Journal of clinical oncology 31.3 (2013): 344-350.

Jackson, Robert J., et al. "Limitations of stereotactic biopsy in the initial management of gliomas." Neuro-oncology 3.3 (2001): 193-200.

McGirt, Matthew J., et al. "Independent predictors of morbidity after image-guided stereotactic brain biopsy: a risk assessment of 270 cases." *Journal of neurosurgery* 102.5 (2005): 897-901.

Mitra, Sushmita, and B. Uma Shankar. "Medical image analysis for cancer management in natural computing framework." Information Sciences 306 (2015): 111-131.