ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΣΧΟΛΗ ΚΟΙΝΩΝΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΓΕΩΓΡΑΦΙΑΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

ΕΦΑΡΜΟΓΗ ΝΕΥΡΩΝΙΚΟΥ ΔΙΚΤΥΟΥ ΣΤΗ ΤΑΥΤΟΠΟΙΗΣΗ ΚΑΙ ΧΑΡΤΟΓΡΑΦΗΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΧΡΗΣΗ ΜΗ ΕΠΑΝΔΡΩΜΕΝΟΥ ΑΕΡΟΣΚΑΦΟΥΣ

ΜΑΜΑΤΣΑΣ ΔΗΜΗΤΡΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΗΜΗΤΡΙΟΣ ΚΑΒΡΟΥΔΑΚΗΣ

ΜΥΤΙΛΗΝΗ, 2019

POSTGRADUATE DISSERTATION

NEURAL NETWORK IMPLEMENTATION IN THE IDENTIFICATION AND MAPPING OF OBJECTS USING UNMANNED AIRCRAFT VIHICLE

MAMATSAS DIMITRIOS

SUPERVISOR: DR. DIMITRIS KAVROUDAKIS

MYTILENE, 2019

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή με τίτλο «Εφαρμογή νευρωνικού δικτύου στη ταυτοποίηση και χαρτογράφηση αντικειμένων με χρήση Μη επανδρωμένου αεροσκάφους» έχει σκοπό τη διαδικασία εκπαίδευσης ενός συνελικτικού νευρωνικού δικτύου, το οποίο αναλαμβάνει τον ρόλο να πραγματοποιεί την ταυτοποίηση και την χαρτογράφηση της θέσης των αντικειμένων μέσω ροής βίντεο ή απλής φωτογραφίας, σε πραγματικό χρόνο από Μη επανδρωμένο αεροσκάφος (στην περίπτωσή μας drone).

Με συνδυασμό του τροποποιημένου νευρωνικού δικτύου YOLO και της ροής βίντεο από το drone που χρησιμοποιήθηκε, μπορέσαμε να αγγίξουμε μια ταχύτητα επεξεργασίας ροής βίντεο της τάξης των 17-22 fps, ταχύτητα που αγγίζει τον πραγματικό χρόνο, καθώς και να πραγματοποιείτε η καταγραφή της θέσης των αντικειμένων που ανιχνεύονται.

Η εργασία αυτή αποτελείται από 4 κεφάλαια :

Στο **πρώτο** κεφάλαιο, αναλύεται η έννοια και η δομή ενός νευρωνικού δικτύου, τόσο θεωρητικά όσο και μαθηματικά. Γίνεται, επίσης, αναφορά και στα στάδια λειτουργίας ενός νευρωνικού δικτύου.

Στο **δεύτερο** κεφάλαιο, αρχικά περιγράφεται το πώς μεταβήκαμε από την απλή ταξινόμηση των εικόνων στην αναγνώριση αντικειμένων. Στη συνέχεια, γίνεται ανάλυση της διαδικασίας ταξινόμησης.

Στο **τρίτο** κεφάλαιο, αφού αναφερθεί και σχολιασθεί το λογισμικό που χρησιμοποιήθηκε γίνεται ανάλυση όλων των μέσων και του εξοπλισμού που χρησιμοποιήθηκε αλλά και η διαδικασία της εκπαίδευσης και τροποποίησης του νευρωνικού δικτύου.

Στο **τέταρτο** κεφάλαιο, παρουσιάζονται αρχικά συμπεράσματα που προέκυψαν και ολοκληρώνοντας, υπάρχουν προτάσεις για μελλοντική ανάπτυξη και περαιτέρω εφαρμογή του δικτύου.

Λέξεις κλειδί: YOLO, UAV, αναγνώριση, χαρτογράφηση, συνελικτικό νευρωνικό δίκτυο

Abstract

This postgraduate dissertation titled "Neural Network Implementation in the Identification and Mapping of Objects using Unmanned Aircraft Vehicle" aims at the training of a convolutional neural network, which enables the identification and mapping of object location via photography or real-time video stream from an unmanned aircraft vehicle (UAV), in our case a drone.

By combining the modified YOLO neural network and the video stream from the drone, we were able to reach a video stream processing rate of 17-22 fps, a speed near real time, as well as capture the position of the objects that are detected.

This dissertation consists of 4 chapters:

The **first** chapter analyzes the concept and structure of a neural network, both theoretically and mathematically. Reference is also made to the different stages of a neural network.

In the **second** chapter, we first describe how we have switched from simple sorting of images to object recognition. The classification process is then analyzed.

The **third** chapter, after reporting and commenting on the software used, follows an analysis of all the tools and equipment used, as well as the process of education and modification of the neural network we made.

The **fourth** chapter presents the conclusions that have been reached, following proposals for future development and further implementation of the network.

Key words: YOLO, UAV, identification, mapping, CNN

Περιεχόμενα

Πίνακας Εικόνων

Εισαγωγή

Πρόσφατες επιτυχίες μεθόδων βαθιάς εκπαίδευσης με εξειδίκευση στην επίλυση ιδιαίτερα πολύπλοκων ενεργειών από ακατέργαστα δεδομένα που προέρχονται απευθείας από αισθητήρες, έχουν δημιουργήσει έναν ιδιαίτερο ενθουσιασμό στην ερευνητική κοινότητα. Ωστόσο, το συγκεκριμένο αντικείμενο έρευνας δεν αποτελεί νέα τεχνολογία. Ξεκίνησε να χρησιμοποιείται από το 1968, όταν ο Ivakhnenko (Ivakhnenko, 1971) εκπαίδευσε ένα νευρωνικό δίκτυο 8 στρωμάτων χρησιμοποιώντας GMDH[1]. Ο όρος βαθιά εκμάθηση άρχισε να χρησιμοποιείται κατά τη διάρκεια της δεκαετίας του 2000, όταν τα CNNs, ένα μοντέλο υπολογιστικού προτύπου από τη δεκαετία του '80 (Fukushima, 1988) το οποίο εκπαιδεύτηκε αποτελεσματικά στη δεκαετία του '90 (Y. LeCun L. B., 1998), ήταν σε θέση να παρέχει αξιοπρεπή αποτελέσματα σε εργασίες οπτικής αναγνώρισης αντικειμένων. Την εποχή εκείνη, τα σύνολα δεδομένων ήταν μικρά και οι υπολογιστές δεν ήταν αρκετά ισχυροί, έτσι η απόδοση ήταν συχνά παρόμοια ή χειρότερη από εκείνη των κλασικών αλγόριθμων τεχνητής όρασης (Computer Vision). Η ανάπτυξη της CUDA[2] για μονάδες GPU της Nvidia που επέτρεψαν πάνω από 1000 GFLOPS[3] ανά δευτερόλεπτο και τη δημοσίευση του ImageNet, με 1,2 εκατομμύρια εικόνες που ταξινομήθηκαν σε 1000 κατηγορίες (J. Deng W. D., 2009), ήταν σημαντικά γεγονότα για τη διάδοση των πολυεπίπεδων νευρωνικών δικτύων (10^9 έως 10^{10} συνδέσεις και 10^7 έως 10^9 παραμέτρους). Αυτά τα βαθιά μοντέλα έδειξαν εξαιρετική απόδοση όχι μόνο στη τεχνητή όραση αλλά και σε άλλες εργασίες, όπως π.χ. την αναγνώριση ομιλίας, την επεξεργασία σημάτων και την επεξεργασία φυσικής γλώσσας (Y. Bengio A. C., 2013).

[1] Η Ομαδική Μέθοδος Επεξεργασίας Δεδομένων (Group Method of Data Handling) εφαρμόστηκε σε μια μεγάλη ποικιλία τομέων βαθιάς εκμάθησης και ανακάλυψης γνώσεων, πρόβλεψης και εξόρυξης δεδομένων, βελτιστοποίησης και αναγνώρισης προτύπων. Οι επαγωγικοί αλγόριθμοι GMDH δίνουν τη δυνατότητα να εντοπίζονται αυτόματα αλληλεπιδράσεις στα δεδομένα, να επιλέγεται η βέλτιστη δομή του μοντέλου ή του δικτύου και να αυξάνεται η ακρίβεια των υφιστάμενων αλγορίθμων.

[2] Αρχιτεκτονική λογισμικού για διαχείριση δεδομένων σε παράλληλο προγραμματισμό η οποία επιτρέπει την χρήση της/των GPU ως γενικές μονάδες επεξεργασίας.

[3] Οι λειτουργίες κυμαινόμενου σημείου ανά δευτερόλεπτο (Floating Point Operations Per Second: FLOPS, flops ή flop/s) είναι μια μέθοδος μέτρησης της απόδοσης υπολογιστών, χρήσιμο σε πεδία επιστημονικών υπολογισμών που απαιτούν υπολογισμούς μεταβλητού σημείου.

Λόγω της ευελιξίας, των δυνατοτήτων αυτοματισμού και του χαμηλού κόστους των UAV, έχει σημειωθεί μια τεράστια αύξηση στην εφαρμογή τους τα τελευταία χρόνια. Μερικά παραδείγματα περιλαμβάνουν την επιθεώρηση γραμμών ηλεκτρικής ενέργειας (C. Martinez, 2014), την εποπτεία και διαχείριση της άγριας πανίδας (M. A. Olivares-Mendez, 2015), την επιθεώρηση κτιρίων (A. Carrio, 2016) και τη καλλιέργεια γης (L. Li, 2016). Ωστόσο, τα UAV έχουν και περιορισμούς οι οποίοι αφορούν το μέγεθος, το βάρος και τη κατανάλωση ισχύος του φορτίου καθώς και τη περιορισμένη εμβέλεια και αντοχή τους. Αυτοί οι περιορισμοί δεν μπορούν να αγνοηθούν και είναι ιδιαίτερα σημαντικοί παράγοντες για την εφαρμογή που θέλουμε να πραγματοποιήσουμε ιδιαίτερα όταν ο αλγόριθμος βαθιάς εκπαίδευσης πρέπει να λειτουργεί επί του σκάφους UAV πράγμα που σημαίνει ότι θα πρέπει να υπάρχει επιπλέον εξοπλισμός επί του σκάφους.

ΚΕΦΑΛΑΙΟ 1

1. Τεχνητά Νευρωνικά Δίκτυα

1.1 Εισαγωγή

Ένα Τεχνητό Νευρωνικό Δίκτυο (ΤΝΔ) είναι ένα μαθηματικό μοντέλο ή υπολογιστικό μοντέλο βασισμένο στα βιολογικά νευρωνικά δίκτυα. Μπορεί επίσης να παρομοιαστεί και ως ένα προσαρμοστικό στατιστικό μοντέλο σε αναλογία με τη δομή του εγκεφάλου. Το νευρωνικό δίκτυο βρίσκει εφαρμογή στη μοντελοποίηση (modeling), αναγνώριση προτύπων (pattern recognition), καθώς επίσης και την επεξεργασία και έλεγχο σημάτων χάρη και την ικανότητα μάθησης από δεδομένα εισόδου. (Hykin, 1999).

Τα ΤΝΔ έχουν μελετηθεί για περισσότερες από έξι δεκαετίες από όταν ο Rosenblatt εφάρμοσε τα πρώτα επίπεδα μονής αντίληψης σε μοντέλα εκπαίδευσης πρότυπων-ταξινόμησης, στα τέλη της δεκαετίας του 1950 και εισήγαγε τον όρο Αντίληπτρο (*Perceptron*). Πρόσφατα, πολλοί ερευνητές εφαρμόζουν τα νευρωνικά δίκτυα στους τομείς της μηχανικής πληροφορικής και της τεχνητής νοημοσύνης. Έχουν χρησιμοποιηθεί σε μεγάλο αριθμό εφαρμογών και έχουν αποδειχθεί αποτελεσματικά στην εκτέλεση σύνθετων λειτουργιών σε διάφορα πεδία.

Υπάρχουν αρκετοί ορισμοί, όσων αφορά τα νευρωνικά δίκτυα. Ωστόσο, για τους σκοπούς της παρούσας μελέτης θα υιοθετηθεί αυτός που χρησιμοποιήθηκε από τους Aleksander και Morton το 1990 (Morton, 1990):

> Ένα Τεχνητό Νευρωνικό Δίκτυο είναι ένας παράλληλος και κατανεμημένος επεξεργαστής που έχει κατασκευαστεί από απλές μονάδες επεξεργασίας (νευρώνες), έχει μια φυσική κλίση στο να αποθηκεύει εμπειρική γνώση και έχει την ικανότητα να τη χρησιμοποιήσει. Μοιάζει στον ανθρώπινο εγκέφαλο με δύο τρόπους:

1. Η γνώση εισέρχεται στο δίκτυο από το περιβάλλον μέσω μιας διαδικασίας εκπαίδευσης

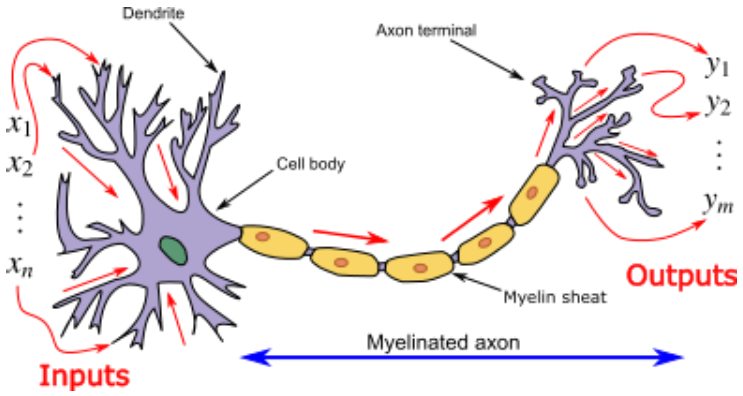
2. Η αποθήκευση της γνώσης γίνεται μέσω των συνάψεων που υπάρχουν στις διασυνδέσεις μεταξύ των νευρώνων και ονομάζονται συναπτικά βάρη.

Η διαδικασία που χρησιμοποιείται για την εκπαίδευση ενός ΤΝΔ καλείται *αλγόριθμος εκπαίδευσης*, και ο σκοπός της είναι να μεταβάλει τα βάρη των διασυνδέσεων του δικτύου έτσι ώστε το δίκτυο να παράγει την επιθυμητή έξοδο. Εκτός από τη μεταβολή των βαρών ενός ΤΝΔ, το δίκτυο έχει την δυνατότητα να μεταβάλει και την τοπολογία του, όπως συμβαίνει στους νευρώνες του ανθρώπινου εγκεφάλου. Δηλαδή, κάποιοι από τους νευρώνες σταματούν να λειτουργούν και πεθαίνουν ενώ κάποιοι άλλοι δημιουργούν νέες συνδέσεις.

1.2. Βαθιά Νευρωνικά Δίκτυα (DNN)

1.2.1 Ο Ανθρώπινος Εγκέφαλος

Τα ΤΝΔ εμπνεύστηκαν από τα βιολογικά ευρήματα που σχετίζονται με τη συμπεριφορά του εγκεφάλου ως ένα δίκτυο κόμβων που ονομάζονται νευρώνες. Ο ανθρώπινος εγκέφαλος εκτιμάται ότι έχει περίπου 10 δισεκατομμύρια νευρώνες, όπου ο καθένας συνδέεται κατά μέσο όρο με 10.000 άλλους νευρώνες. Το νευρωνικό μοντέλο προσομοιώνει τη βασική δομή του ανθρώπινου εγκεφάλου που παρουσιάζεται στην Εικόνα 1. Η δομή των άκρων του μοιάζει με δέντρο και ονομάζονται *Δενδρίτες*. Οι νευρώνες συλλέγουν σήματα από άλλους νευρώνες μέσω αυτών των δενδριτών. Ο νευρώνας στέλνει ηλεκτρικά σήματα μέσω ενός μακρύ και λεπτού αγωγού γνωστού ως *νευρικός άξονας* ή απλά *άξονας*, ο οποίος χωρίζεται σε χιλιάδες κλάδους. Στο τέλος κάθε κλάδου, μια δομή που ονομάζεται σύναψη μετατρέπει τις δραστηριότητες από τον άξονα σε ηλεκτρικά σήματα τα οποία αναστέλλουν ή διεγείρουν τη δραστηριότητα στον συνδεδεμένο νευρώνα. Όταν ένας νευρώνας λαμβάνει ως είσοδο μια διέγερση που είναι αρκετά μεγάλη σε σύγκριση με την ανασταλτική του λειτουργία, στέλνει ένα σήμα ηλεκτρικής δραστηριότητας μέσω του άξονά του. Η εκπαίδευση γίνεται αλλάζοντας την αποτελεσματικότητα των συνάψεων έτσι ώστε να αλλάζουν οι επιδράσεις μεταξύ των νευρώνων.



Εικόνα 1: Μοντέλο βιολογικού νευρώνα

Το μοντέλο του βιολογικού νευρώνα είναι το θεμέλιο ενός τεχνητού νευρώνα. Προσομοιώνει τα τέσσερα βασικά συστατικά του φυσικού νευρώνα: τον δενδρίτη, το σώμα, τον άξονα και τις συνάψεις.

Στη βιβλιογραφία των νευρωνικών δικτύων, η μονάδα που αναλογεί με τον βιολογικό νευρώνα αναφέρεται ως στοιχείο επεξεργασίας [Processing Elements: PE] ή μονάδες [Units]. Ένα στοιχείο επεξεργασίας έχει πολλές διαδρομές εισόδου και συνδυάζει, συνήθως με ένα απλό άθροισμα, τις τιμές αυτών των διαδρομών εισαγωγής για να εξάγει ένα και μόνο αποτέλεσμα.

1.2.2 Μοντέλο Νευρώνα

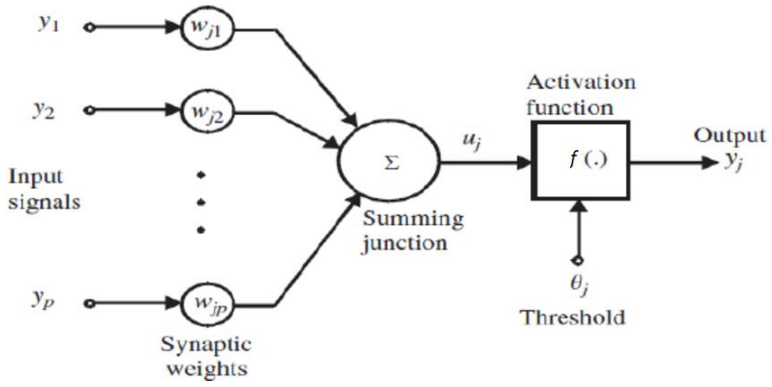
Κατά τη δημιουργία ενός λειτουργικού μοντέλου του βιολογικού νευρώνα, υπάρχουν τρία βασικά συστατικά:

1. Ένα σύνολο από συνάψεις (διασυνδέσεις), η κάθε μια από τις οποίες χαρακτηρίζεται από κάποιο βάρος. Συγκεκριμένα, ένα σήμα y_j στην είσοδο μιας σύναψης j που είναι συνδεδεμένη με το νευρώνα k πολλαπλασιάζεται με το βάρος της σύναψης w_{ji}. Οι υποδείκτες του w έχουν την εξής σημασία: Ο πρώτος υποδείκτης αναφέρεται στον εν λόγω νευρώνα και ο δεύτερος στην είσοδο της σύναψης όπου αναφέρεται το βάρος. Το βάρος w_{ji} είναι θετικό όταν η σχετική σύναψη διεγείρει το νευρώνα και αρνητικό όταν η σύναψη είναι ανασταλτική.

2. Έναν αθροιστή που αθροίζει τα εισερχόμενα σήματα στον νευρώνα τα οποία έχουν πολλαπλασιαστεί με το βάρος της αντίστοιχης σύναψης από την οποία εισήλθαν. Οι διαδικασίες που περιγράφονται εδώ αποτελούν ένα γραμμικό συνδιαστή.

3. Μια συνάρτηση ενεργοποίησης για τον περιορισμό του μεγέθους της εξόδου ενός νευρώνα. Συνήθως το κανονικοποιημένο εύρος της εξόδου ενός νευρώνα είναι το κλειστό σύνολο [0,1] ή [-1,1].

Η αρχιτεκτονική ενός τεχνητού νευρώνα φαίνεται στην παρακάτω Εικόνα 2.



Εικόνα 2: Μοντέλο τεχνητού νευρώνα k

Η εσωτερική δραστηριότητα του παραπάνω μοντέλου περιγράφεται με την παρακάτω εξίσωση:

$$u_j = \sum_{j=1}^{p} \mathrm{w_{kj}y_j}$$

Εξίσωση 1: Εξίσωση αθροίσματος βαρών για νευρωνικά δίκτυα

όπου $y_{1,}\, y_2, \ldots, y_p$είναι τα εισερχόμενα σήματα, $w_{j1}, w_{j2}, \ldots, w_{jp}$ είναι τα βάρη των συνάψεων του νευρώνα k, u_j είναι η έξοδος του γραμμικού συνδιαστή, $f(\bullet)$ είναι η συνάρτηση ενεργοποίησης και y_j είναι το σήμα που δίνει ως έξοδο ο νευρώνας.

Η έξοδος του νευρώνα, θα είναι επομένως το αποτέλεσμα κάποιας συνάρτησης ενεργοποίησης της τιμής u_j.

1.2.2.1 Τα Μέρη ενός Νευρώνα

Ένας τεχνητός νευρώνας αποτελείται από επτά κύρια μέρη: *Συντελεστή βαρύτητας*, *αθροιστή, συνάρτηση μεταφοράς, κλιμάκωση και περιορισμό, λειτουργία εξόδου, λειτουργία και τιμές λειτουργίας σφάλματος* και τέλος, *λειτουργία εκπαίδευσης* (Anderson D. and McNeil, 1992). Αυτά τα στοιχεία ενεργοποιούνται κάθε φορά που ο νευρώνας χρησιμοποιείται σε οποιοδήποτε επίπεδο: στην είσοδο, κατά την έξοδο ή ως ένα από τα κρυφά επίπεδα.

1.2.2.1.1 *Συντελεστής Βαρύτητας*

Ένας νευρώνας συνήθως λαμβάνει ταυτόχρονα πολλές εισροές. Κάθε είσοδος έχει το δικό της σχετικό βάρος, το οποίο δίνει στην είσοδο το βάρος που χρειάζεται για την άθροιση του στοιχείου επεξεργασίας. Τα βάρη λειτουργούν ως προσαρμοστικοί συντελεστές εντός του δικτύου και καθορίζουν την ένταση του σήματος εισόδου στον τεχνητό νευρώνα. Λειτουργούν ουσιαστικά ως ένα μέσο μέτρησης της δύναμης μιας εισόδου. Αυτή η δύναμη μπορεί να τροποποιηθεί σύμφωνα με τα διάφορα σύνολα εκπαίδευσης και μέσω των κανόνων εκπαίδευσης.

1.2.2.1.2 *Αθροιστής*

Το πρώτο βήμα στη λειτουργία του στοιχείου επεξεργασίας είναι να υπολογισθεί το σταθμισμένο άθροισμα όλων των εισροών. Μαθηματικά, οι είσοδοι και το αντίστοιχο βάρος είναι διανύσματα που μπορούν να αναπαρασταθούν ως $(y_1, y_2, \dots, y_p)$ και $(w_{j1}, w_{j2}, \dots, w_{jp})$ αντίστοιχα. Το συνολικό σήμα εισόδου ή το εσωτερικό προϊόν αυτών δύο διανυσμάτων είναι το αποτέλεσμα $f(\bullet)$. Το αποτέλεσμα είναι ένας αριθμός, και όχι ένα διάνυσμα πολλαπλών στοιχείων. Ο αθροιστής μπορεί να έχει μια πιο σύνθετη μορφή από το να δέχεται απλώς μια είσοδο και να υπολογίζει το άθροισμα των βαρών. Οι συντελεστές εισόδου και βάρους μπορούν να συνδυαστούν με πολλούς διαφορετικούς τρόπους πριν να περάσουν στη συνάρτηση μεταφοράς.

1.2.2.1.3 *Συνάρτηση Μεταφοράς*

Το αποτέλεσμα του αθροιστή, σχεδόν πάντα είναι το σταθμισμένο άθροισμα των εισόδων, μεταφέρεται σε μια έξοδο μέσω μιας αλγοριθμικής διαδικασίας γνωστή ως συνάρτηση

μεταφοράς. Στη συνάρτηση μεταφοράς, το άθροισμα μπορεί να συγκριθεί με κάποιες συγκεκριμένες τιμές εισόδου για το προσδιορισμό της. Εάν το άθροισμα είναι μεγαλύτερο από τη τιμή εισόδου, τα στοιχεία επεξεργασίας δημιουργούν ένα σήμα. Αν το άθροισμα είναι μικρότερο από το όριο, δε παράγεται σήμα. Και οι δύο τύποι απόκρισης είναι σημαντικοί.

1.2.2.1.4 *Κλιμάκωση και Περιορισμός*

Μόλις ολοκληρωθεί η συνάρτηση μεταφοράς, το αποτέλεσμα μπορεί να περάσει από πρόσθετη επεξεργασία όπως είναι αυτή της κλιμάκωσης και περιορισμού. Αυτή η κλιμάκωση πολλαπλασιάζει τον συντελεστή κλίμακας με την τιμή μεταφοράς και στη συνέχεια προσθέτει μια *μετατόπιση*. Ο περιορισμός είναι ο μηχανισμός, ο οποίος διασφαλίζει ότι το κλιμακωτό αποτέλεσμα δεν υπερβαίνει ένα ανώτερο ή κατώτερο όριο ([0,1] ή [-1,1]). Αυτός ο τύπος κλιμάκωσης και περιορισμού χρησιμοποιείται κυρίως σε τοπολογίες για τη δοκιμή μοντέλων βιολογικών νευρώνων.

1.2.2.1.5 *Λειτουργία Εξόδου*

Κάθε στοιχείο επεξεργασίας επιτρέπεται να εξάγει ένα και μοναδικό σήμα, το οποίο μπορεί να αποτελέσει είσοδο σε εκατοντάδες άλλους νευρώνες. Έτσι ακριβώς λειτουργεί και ένας βιολογικός νευρώνας, όπου υπάρχουν πολλές εισροές και μόνο μία έξοδος. Κανονικά, η έξοδος είναι άμεσα ισοδύναμη με το αποτέλεσμα της λειτουργίας μεταφοράς. Ορισμένες τοπολογίες δικτύου, ωστόσο, τροποποιούν το αποτέλεσμα της συνάρτησης μεταφοράς, ώστε η τιμή να μπορεί να συνεργαστεί με τις τιμές γειτονικών στοιχείων επεξεργασίας. Οι νευρώνες επιτρέπεται να ανταγωνίζονται μεταξύ τους, αναστέλλοντας στοιχεία επεξεργασίας εκτός αν αυτοί έχουν μεγάλη σημασία για το σύστημα.

1.2.2.1.6 *Λειτουργία και Τιμές Σφάλματος*

Στα περισσότερα δίκτυα εκπαίδευσης το σφάλμα του τεχνητού νευρώνα υπολογίζεται ως η διαφορά μεταξύ της τρέχουσας εξόδου και της επιθυμητής εξόδου. Αυτό το ακατέργαστο σφάλμα στη συνέχεια μεταφέρεται από τη συνάρτηση σφάλματος και επανεισάγεται στον νευρώνα ώστε να ταιριάζει με την αρχιτεκτονική του συγκεκριμένου δικτύου. Έπειτα τα

σφάλματα μεταφέρονται σε λειτουργίες εκπαίδευσης ενός άλλου στοιχείου επεξεργασίας. Αυτός ο όρος σφάλματος μερικές φορές ονομάζεται *τρέχον σφάλμα*.

1.2.2.1.7 *Λειτουργία Εκπαίδευσης*

Ο σκοπός της λειτουργίας εκπαίδευσης είναι να τροποποιήσει τα μεταβλητά βάρη σύνδεσης στις εισόδους του κάθε στοιχείου επεξεργασίας σύμφωνα με ορισμένους νευρωνικούς αλγορίθμους. Αυτή η διαδικασία αλλαγής του βάρους των συνδέσεων εισόδου για την επίτευξη κάποιου επιθυμητού αποτελέσματος θα μπορούσε επίσης να ονομάζεται και λειτουργία προσαρμογής.

1.2.3 Τύποι Συναρτήσεων Ενεργοποίησης

Η συνάρτηση ενεργοποίησης *f(●)* εκτελεί μια μαθηματική λειτουργία. Όπως προαναφέρθηκε, η συνάρτηση ενεργοποίησης ενεργεί ως λειτουργία συμμόρφωσης, έτσι ώστε η έξοδος ενός νευρώνα σε ένα νευρωνικό δίκτυο να είναι μεταξύ ορισμένων τιμών. Σε γενικές γραμμές, συναντώνται κυρίως τρεις διαφορετικές συναρτήσεις ενεργοποίησης:

1. Η συνάρτηση κατώφλι (*Threshold function*), γνωστή και ως συνάρτηση Heaviside, η οποία παίρνει τη τιμή 0 εάν ο αθροιστής είναι μικρότερος από την τιμή κατώφλι *ν* (*threshold*) και τη τιμή 1 όταν η τιμή είναι μεγαλύτερη ή ίση με τη τιμή κατώφλι που έχει ορισθεί. Αυτό το είδος συνάρτησης ενεργοποίησης ορίζεται από το παρακάτω τύπο:

$$f(\nu) = \begin{cases} 0, & \varepsilon\acute{\alpha}\nu\ \nu < 0 \\ 1, & \varepsilon\acute{\alpha}\nu\ \nu \geq 0 \end{cases}$$

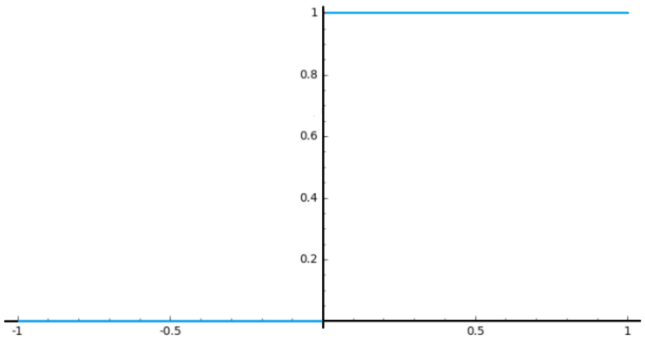
Εξίσωση 2: Συνάρτηση Κατώφλι

Αντίστοιχα, η έξοδος του νευρώνα k με τη χρήση μιας τέτοιας συνάρτησης εκφράζεται ως:

$$y_k = \begin{cases} 0, & \varepsilon\acute{\alpha}\nu\ u_k < 0 \\ 1, & \varepsilon\acute{\alpha}\nu\ u_k \geq 0 \end{cases}$$

Εξίσωση 3: Τιμές εξόδου νευρώνα σύμφωνα με τη Συνάρτηση Κατώφλι

Σε αυτό το μοντέλο η έξοδος του νευρώνα παίρνει τη τιμή 1 αν η ολική δυνατότητα ενεργοποίησης του νευρώνα είναι μη αρνητική διαφορετικά παίρνει τη τιμή 0. Η μορφή της συνάρτησης φαίνεται παρακάτω:



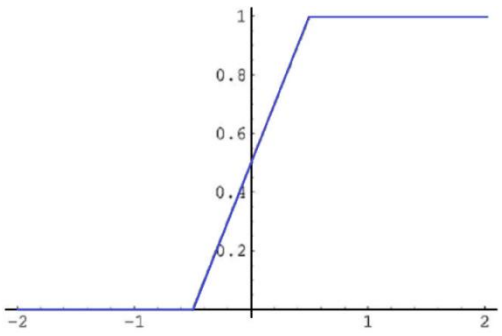
Εικόνα 3: Γράφημα της συνάρτησης ενεργοποίησης κατώφλι

2. Τμηματική γραμμική συνάρτηση (*Piecewise-Linear Function*). Αυτή η λειτουργία μπορεί να πάρει και πάλι τις τιμές 0 ή 1, αλλά μπορεί επίσης να πάρει και ενδιάμεσες τιμές οι οποίες εξαρτώνται από τον παράγοντα ενίσχυσης (*amplification factor*), όπου ο παράγοντας ενίσχυσης μέσα στη γραμμική περιοχή της συνάρτησης θεωρείται μονάδα. Αυτό το είδος συνάρτησης ενεργοποίησης ορίζεται από τον τύπο:

$$f(\nu) = \begin{cases} 1, \text{εάν } \nu \geq \frac{1}{2} \\ \nu, \text{εάν } -\frac{1}{2} > \nu > \frac{1}{2} \\ 0, \text{εάν } \nu \leq \frac{1}{2} \end{cases}$$

Εξίσωση 4: Τμηματική Γραμμική Συνάρτηση

και η μορφή της είναι η παρακάτω:



Εικόνα 4: Γράφημα της τμηματικής γραμμικής συνάρτησης

3. Σιγμοειδής συνάρτηση ενεργοποίησης (*Sigmoid Function*). Η σιγμοειδής συνάρτηση είναι η συνηθέστερη συνάρτηση ενεργοποίησης για τη κατασκευή τεχνητών νευρωνικών δικτύων (συνήθως χρησιμοποιείται στην έξοδο του νευρώνα ως κρυφό επίπεδο) και είναι μια ειδική περίπτωση της συνάρτησης ενεργοποίησης που έχει μια χαρακτηριστική καμπύλη σχήματος "S". Η λογική συνάρτηση ορίζεται από το τύπο:

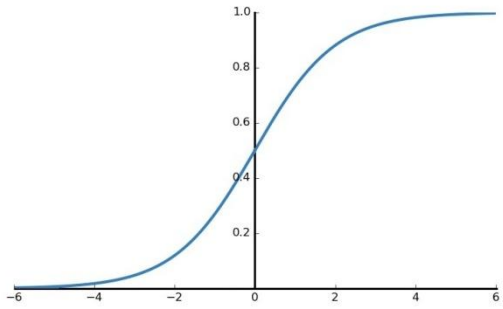
$$f(\nu) = \frac{L}{1+e^{-k(x-x_0)}}$$

Εξίσωση 5: Σιγμοειδής συνάρτηση

όπου το e είναι η βάση φυσικού λογαρίθμου [γνωστό και ως αριθμός Όιλερ (Euler)], x_0 είναι η μέση τιμή της σιγμοειδούς συνάρτησης, το L είναι η μέγιστη τιμή της καμπύλης, και το k είναι η κλίση της καμπύλης. Θέτοντας L=1, k=1, και x_0=0, έχουμε:

$$f(\nu) = \frac{1}{1+e^{-x}}$$

Εξίσωση 6: Τροποποιημένη σιγμοειδής συνάρτηση



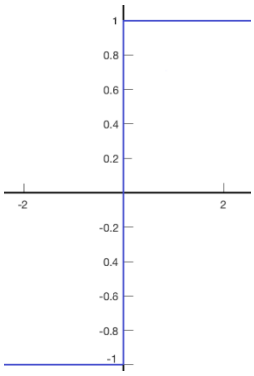
Εικόνα 5: Γράφημα σιγμοειδούς συνάρτηση

Μέχρι τώρα οι συναρτήσεις που αναφέρθηκαν παίρνουν τιμές [0,1], κάποιες φορές όμως είναι επιθυμητό η συνάρτηση ενεργοποίησης να παίρνει τιμές [-1,1] (Sarles, 1997). Σε αυτή

τη περίπτωση η συνάρτηση ενεργοποίησης παίρνει μια αντισυμμετρική μορφή ως προς την αρχή των αξόνων. Συγκεκριμένα η συνάρτηση κατώφλι γίνεται:

$$f(\nu) = \begin{cases} 1, & \text{εάν } \nu > 0 \\ 0, & \text{εάν } \nu = 0 \\ -1, & \text{εάν } \nu < 0 \end{cases}$$

Εξίσωση 7: Τροποποιημένη συνάρτηση κατώφλι



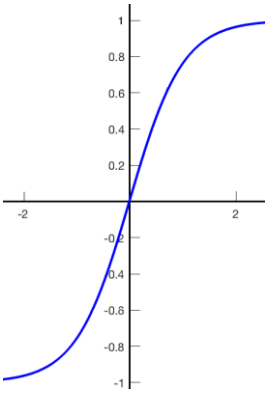
Εικόνα 6: Γράφημα τροποποιημένης συνάρτησης κατώφλι

Ενώ για την σιγμοειδή συνάρτηση μπορούμε να πάρουμε την υπερβολική εφαπτομένη που δίνεται από το παρακάτω τύπο:

$$f(\nu) = \tanh(\nu)$$

Εξίσωση 8: Υπερβολική εφαπτομένη σιγμοειδής συνάρτηση

Και το γράφημα της Εξίσωσης 8 θα είναι της μορφής:



Εικόνα 7: Γράφημα τροποποιημένης σιγμοειδούς συνάρτησης

1.2.3 Η Αρχιτεκτονική των Τεχνητών Νευρωνικών Δικτύων

Τα ΤΝΔ έχουν απλές δομές και έχουν σχεδιαστεί για να μιμούνται τη λειτουργία ενός βιολογικού νευρώνα. Η κύρια σημασία ενός νευρωνικού δικτύου έγκειται στις διασυνδέσεις τους. Οι φυσικές συνδέσεις ενός ανθρώπινου εγκεφάλου είναι πολύ περίπλοκες για να εφαρμοσθούν άμεσα σε ένα τεχνητό νευρωνικό δίκτυο. Οι δομές που έχουν μελετηθεί μέχρι τώρα είναι απλές και είναι σχετικά εύκολες στο να εφαρμοστούν σε τεχνητά νευρωνικά δίκτυα.

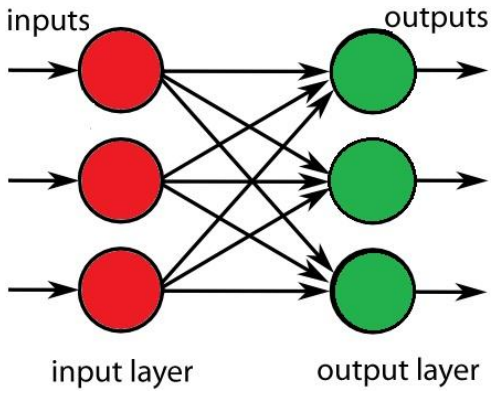
Η αρχιτεκτονική ενός νευρωνικού δικτύου είναι η συγκεκριμένη διάταξη και σύνδεση των νευρώνων που οργανώνονται σε μορφή επιπέδων, αυτή η δομή αποτελεί το δίκτυο. Ορίζεται από μια σειρά επιπέδων, έναν αριθμό κόμβων ανά επίπεδο και το μοτίβο διασύνδεσης μεταξύ των επιπέδων. Γενικά, τα ΤΝΔ παρουσιάζουν παρόμοια δομή, όσων αφορά τα επίπεδα από τα οποία αποτελούνται, με μικρές διαφοροποιήσεις. Υπάρχουν τρεις βασικοί τύποι επιπέδων τα οποία αποτελούνται από ομάδες νευρώνων:

1. Επίπεδα εισόδου. Τα επίπεδα εισόδου αποτελούνται από νευρώνες που λαμβάνουν εισροές από το εξωτερικό περιβάλλον.
2. Επίπεδα εξόδου. Τα επίπεδα εξόδου αποτελούνται από νευρώνες που μεταδίδουν την έξοδο του συστήματος στο χρήστη ή το εξωτερικό περιβάλλον.
3. Ένα ή περισσότερα κρυφά επίπεδα μεταξύ των επιπέδων εισόδου και εξόδου.

Όταν το επίπεδο εισόδου λαμβάνει μια είσοδο, οι νευρώνες του επιπέδου αυτού παράγουν μια έξοδο η οποία εισάγεται στο επόμενο επίπεδο του συστήματος. Ο τρόπος με τον οποίο δομείται το νευρωνικό δίκτυο συνδέεται εσωτερικά με τον αλγόριθμο εκπαίδευσης που χρησιμοποιείται για την εκπαίδευση του δικτύου. Γενικά υπάρχουν τρεις βασικές κατηγορίες αρχιτεκτονικής νευρωνικών δικτύων (Hykin, 1999), τα δίκτυα εμπρόσθιας τροφοδοσίας μονού επιπέδου, δίκτυα εμπρόσθιας τροφοδοσίας πολλαπλών επιπέδων και τα αναδραστικά νευρωνικά δίκτυα.

1.2.3.1 Δίκτυα Εμπρόσθιας Τροφοδότησης Ενός Επιπέδου (Single-layer Feedforward Networks)

Ένα δίκτυο εμπρόσθιας τροφοδότησης είναι ένα δίκτυο όπου οι νευρώνες στο πρώτο επίπεδο στέλνουν την έξοδό τους στους νευρώνες του δευτέρου επιπέδου, αλλά χωρίς να υπάρχει η δυνατότητα ανατροφοδότησής τους από τους νευρώνες του δευτέρου επιπέδου. Ένα δίκτυο το οποίο διαθέτει μόνο επίπεδα εισόδου και εξόδου ονομάζεται *δίκτυο μονού επιπέδου(single layered network).* Με τον όρο μονό επίπεδο εννοούμε το επίπεδο εξόδου που περιέχει και τους νευρώνες όπου γίνονται οι υπολογισμοί. Αξίζει να σημειωθεί ότι δεν υπολογίζεται το επίπεδο εισόδου καθώς δε γίνονται καθόλου υπολογισμοί σε αυτό.



Εικόνα 8: Τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης ενός επιπέδου

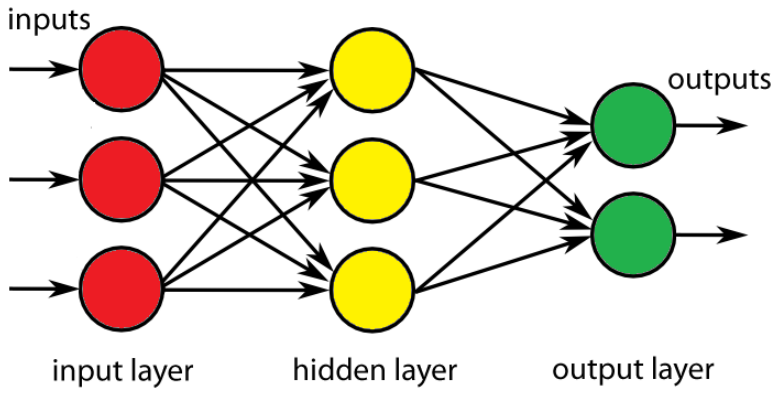
1.2.3.2 Δίκτυα Εμπρόσθιας Τροφοδότησης Πολλαπλών Επιπέδων (Multilayer Feedforward Networks)

Ένα δίκτυο εμπρόσθιας τροφοδότησης πολλαπλών επιπέδων αποτελείται από ένα επίπεδο κόμβων εισόδου, ένα ή περισσότερα κρυφά επίπεδα κόμβων και από ένα επίπεδο κόμβων εξόδου. Αυτή η δομή ονομάζεται *πολυστρωματική (multilayer)* επειδή διαθέτει ένα επίπεδο κόμβων επεξεργασίας (κρυφοί κόμβοι) εκτός από το επίπεδο με τους κόμβους εξόδου. Αυτά τα δίκτυα ονομάζονται *εμπρόσθιας τροφοδότησης (feedforward)* επειδή η έξοδος από ένα επίπεδο νευρώνων τροφοδοτείται προς τα εμπρός, προς το επόμενο επίπεδο νευρώνων. Δεν υπάρχουν συνδέσεις προς τα πίσω και οι συνδέσεις δεν παραβλέπουν ποτέ ένα επίπεδο. Συνήθως, τα επίπεδα είναι πλήρως συνδεδεμένα, πράγμα που σημαίνει ότι όλες οι μονάδες του ενός επιπέδου συνδέονται με όλες τις μονάδες του

επόμενου επιπέδου. Αυτό σημαίνει ότι όλοι οι κόμβοι εισόδου συνδέονται με όλους τους κόμβους στο κρυφό επίπεδο και όλες οι μονάδες στο κρυφό επίπεδο είναι συνδεδεμένες με όλες τις μονάδες εξόδου.

Συνήθως, ο προσδιορισμός του αριθμού μονάδων εισόδου και μονάδων εξόδου καθορίζεται από το πεδίο στο οποίο θα εφαρμόσουμε το δίκτυο. Ωστόσο, ο προσδιορισμός του αριθμού των κρυφών μονάδων είναι ένα κομμάτι το οποίο απαιτεί εμπειρία και πειραματισμό για το προσδιορισμό του βέλτιστου αριθμού κρυφών μονάδων. Πολύ λίγες κρυφές μονάδες θα εμποδίσουν το δίκτυο να είναι σε θέση να μάθει την απαιτούμενη συνάρτηση, επειδή θα έχει πολύ λίγους βαθμούς ελευθερίας. Πάρα πολλές κρυφές μονάδες μπορεί να προκαλέσουν στο δίκτυο μια υπεραπλούστευση των δεδομένων εκπαίδευσης, μειώνοντας έτσι την ακρίβεια των αποτελεσμάτων. Πολλές κρυφές μονάδες μπορούν επίσης να αυξήσουν σημαντικά το χρόνο εκπαίδευσης.

Στην παρακάτω Εικόνα 9 απεικονίζεται ένα τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης πολλαπλών επιπέδων με ένα κρυφό επίπεδο.

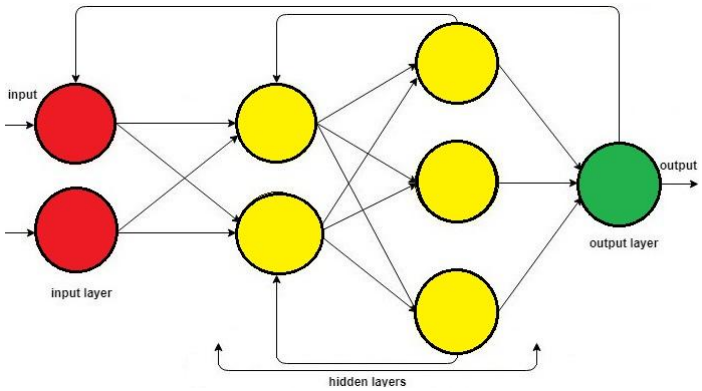


Εικόνα 9: Τεχνητό νευρωνικό δίκτυο εμπρόσθιας τροφοδότησης πολλαπλών επιπέδων με ένα κρυφό επίπεδο

Το νευρωνικό δίκτυο της Εικόνας 9 ονομάζεται πλήρως διασυνδεδεμένο (*fully connected*), καθώς κάθε κόμβος σε κάθε επίπεδο του δικτύου είναι συνδεδεμένος με κάθε κόμβο του επόμενου επιπέδου του δικτύου. Στη περίπτωση που κάποιες από τις συνδέσεις δεν υπάρχουν τότε λέμε ότι το δίκτυο είναι μερικώς διασυνδεδεμένο (partially connected).

1.2.3.3 Αναδραστικά Νευρωνικά Δίκτυα (Recurrent Neural Networks (RNN))

Υπάρχουν αρκετοί τύποι αναδραστικών νευρωνικών δικτύων, οι οποίοι έχουν βρόχους ανατροφοδότησης από το επίπεδο εξόδου στο επίπεδο εισόδου ή από τα κρυφά επίπεδα προς τα επίπεδα εισόδου. Με άλλα λόγια, η αρχιτεκτονική των αναδραστικών δικτύων διαφέρει από την αρχιτεκτονική των δικτύων feedforward καθώς διαθέτει τουλάχιστον ένα βρόχο ανατροφοδότησης. Μια τυπική δομή αρχιτεκτονικής αναδραστικού νευρωνικού δικτύου φαίνεται στην Εικόνα 10, το οποίο έχει βρόχους ανατροφοδότησης από τους νευρώνες εξόδου προς τους νευρώνες του κρυφού επιπέδου αλλά και προς τους νευρώνες εισόδου. Η ύπαρξη ενός βρόχου ανατροφοδότησης έχει επίδραση στη δυνατότητα εκπαίδευσης του δικτύου καθώς και στις επιδόσεις του. Επειδή τα νευρωνικά δίκτυα ανατροφοδότησης διαθέτουν ρυθμιζόμενα βάρη, η κατάσταση του νευρώνα εξαρτάται όχι μόνο από τον όγκο των σημάτων εισόδου, αλλά και από τις προηγούμενες εισόδους του νευρώνα. Το πλεονέκτημα των νευρωνικών δικτύων δυναμικής ανάδρασης είναι ότι είναι σε θέση να μειώσουν αποτελεσματικά τον όγκο των απαιτούμενων δεδομένων εισόδου πράγμα το οποίο συνεπάγεται μικρότερο χρόνο εκπαίδευσης. Ωστόσο, λόγω του μη γραμμικού χαρακτήρα αυτών των δικτύων και των ρυθμιζόμενων βαρών που εφαρμόζουν, δεν είναι εύκολο να εξακριβωθεί η σταθερότητα του δικτύου (Perlovsky, 2001).

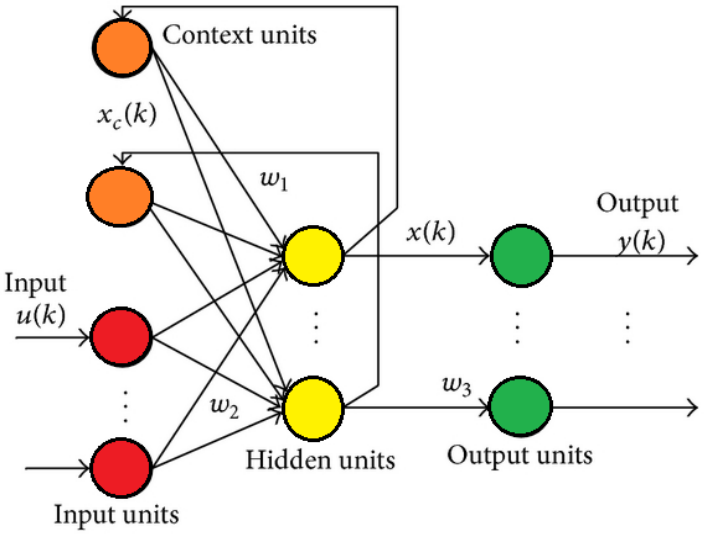


Εικόνα 10: Αναδραστικό τεχνητό νευρωνικό δίκτυο

Τα αναδραστικά νευρωνικά δίκτυα χρησιμοποιούνται για τη πρόβλεψη χρονοσειρών, δηλαδή για την αναγνώριση ακολουθιών όπου υπάρχει σειριακή συνοχή μεταξύ των στοιχείων εισόδου. Αυτό συμβαίνει γιατί τα αναδραστικά νευρωνικά δίκτυα έχουν δυναμική μνήμη η οποία οφείλεται στα επίπεδα ανατροφοδότησης που περιέχουν. Η πρώτη πειραματική επίδειξη (Hykin, 1999) για το αν τα αναδραστικά νευρωνικά δίκτυα

μπορούν να μάθουν τα ενδεχόμενα που μπορούν να προκύψουν από πεπερασμένα δεδομένα έγινε το 1989 πάνω σε συμβολοσειρές. Πιο συγκεκριμένα το δίκτυο έπαιρνε σαν εισόδους συμβολοσειρές και καλούνταν να προβλέψει το επόμενο γράμμα σε ένα δεδομένο λεξιλόγιο συμβολοσειρών που είχε εκπαιδευτεί. Η δυσκολία σε αυτή τη διαδικασία είναι ότι στο λεξιλόγιο κάποιο γράμμα ή κάποια ακολουθία γραμμάτων μπορεί να ακολουθείται από διαφορετικά γράμματα. Το δίκτυο αποδείχτηκε ότι μπορεί να αναπτύξει κάποια αντίληψη για τη σειρά των γραμμάτων στις συμβολοσειρές πάνω στις οποίες είχε διδαχθεί. Υπάρχουν διάφορες αρχιτεκτονικές αναδραστικών νευρωνικών δικτύων, αλλά στα πλαίσια των απαιτήσεων της παρούσας διατριβής θα αρκεστούμε στην περιγραφή της αρχιτεκτονικής του απλού αναδραστικού δικτύου (*simple recurrent network (SRNN)*). Αυτό το μοντέλο είναι μία ειδική περίπτωση μίας κλάσης αναδραστικών νευρωνικών δικτύων που παρουσιάστηκε από τον Elman (Elman, 1990).

Το απλό αναδραστικό νευρωνικό δίκτυο (ή δίκτυο Elman) έχει ένα κρυφό επίπεδο νευρώνων του οποίου η έξοδος συνδέεται αναδραστικά με τις μονάδες πλαισίου (*context units*), ένα επιπλέον επίπεδο νευρώνων το οποίο αποτελείται από μονάδες όπως στην Εικόνα 11.



Εικόνα 11: Απλό αναδραστικό νευρωνικό δίκτυο

1.2.4 Διαδικασία Εκπαίδευσης Νευρωνικού Δικτύου

Η διαδικασία προσαρμογής των βαρών ώστε το δίκτυο να είναι σε θέση να μάθει τη σχέση μεταξύ της εισόδου και του επιθυμητού αποτελέσματος ονομάζεται εκπαίδευση. Η σύνδεση μεταξύ των νευρώνων σε ένα νευρωνικό δίκτυο ονομάζεται ρυθμιζόμενο βάρος ή μέτρο σπουδαιότητας. Σύμφωνα με τον ορισμό των Mendal και McClaren (Hykin, 1999), η εκπαίδευση νευρωνικών δικτύων ορίζεται ως:

> *«Η διαδικασία με την οποία οι ελεύθερες παράμετροι ενός δικτύου προσαρμόζονται μέσω μιας διαδικασίας λήψης ερεθισμάτων από το περιβάλλον στο οποίο είναι ενσωματωμένο. Ο τύπος εκπαίδευσης καθορίζεται από τον τρόπο με τον οποίο πραγματοποιούνται οι αλλαγές στις παραμέτρους.»*

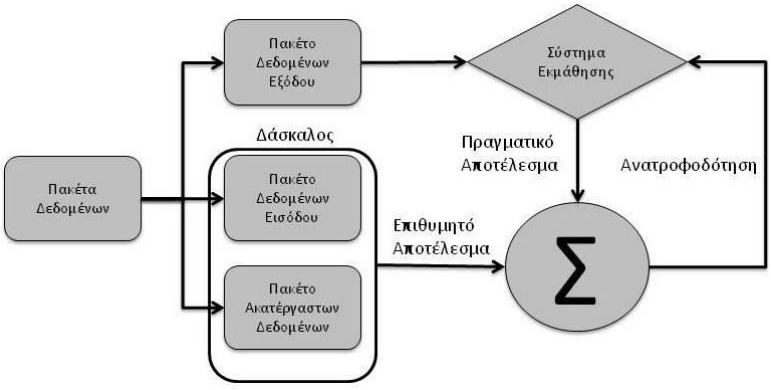
Η διαδικασία για να είναι σε θέση ένα δίκτυο να μάθει το πώς να επιλύει ένα πρόβλημα ακολουθεί μια σειρά από συγκεκριμένους κανόνες, το σύνολο των κανόνων αυτών ονομάζονται *αλγόριθμοι εκπαίδευσης*. Έχουν επινοηθεί διάφοροι αλγόριθμοι εκπαίδευσης ώστε να είναι δυνατός ο υπολογισμός του βέλτιστου συνόλου βαρών που θα οδηγήσουν στην επίλυση του προβλήματος. Σε γενικές γραμμές, όλες οι μέθοδοι εκπαίδευσης ενός αναδραστικού δικτύου μπορούν να ταξινομηθούν σε δύο μεγάλες κατηγορίες, μάθηση με επίβλεψη και μάθηση χωρίς επίβλεψη.

1.2.4.1 Μάθηση με επίβλεψη (Supervised Learning)

Στη μάθηση με επίβλεψη τα νευρωνικά δίκτυα εκπαιδεύονται για να παράγουν τα επιθυμητά αποτελέσματα τα οποία εξαρτώνται από τα δείγματα εισόδου, καθιστώντας τα κατάλληλα για τη μοντελοποίηση και τον έλεγχο δυναμικών συστημάτων, τη ταξινόμηση ηχητικών δεδομένων και για τη πρόβλεψη μελλοντικών γεγονότων. Συνήθως, στα νευρωνικά δίκτυα εφαρμόζεται η τεχνική εκπαίδευσης, μάθηση με επίβλεψη. Υπάρχει ένα πακέτο δεδομένων με παραδείγματα εισόδου και τα αντίστοιχα αναμενόμενα αποτελέσματα εξόδου. Αυτά τα ζεύγη εισόδου-εξόδου παρέχονται από ένα εξωτερικό σύνολο που ονομάζεται *δάσκαλος*.

Η πραγματική έξοδος του δικτύου μπορεί να ταιριάζει με την επιθυμητή έξοδο αλλά μπορεί και όχι καθώς το αποτέλεσμα εξαρτάται από το βάρος στη συγκεκριμένη στιγμή. Σε όλες τις περιπτώσεις οι αλγόριθμοι κατάρτισης τροποποιούν τα βάρη με την ιδέα ότι την επόμενη φορά που το δίκτυο θα δεχθεί ως είσοδο το ίδιο πρότυπο εισόδου, θα έχει ως στόχο να πλησιάσει όσο το δυνατών περισσότερο την έξοδο της σε αυτήν που της υποδείχθηκε κατά την διάρκεια εκπαίδευσης. Στη συνέχεια οι παράμετροι του δικτύου προσαρμόζονται ανάλογα με το πρότυπο που χρησιμοποιείται για την εκπαίδευση και το σφάλμα του δικτύου (δηλαδή την διαφορά μεταξύ της επιθυμητής εξόδου και της εξόδου που στη πράξη δίνει το δίκτυο). Η προσαρμογή αυτών των παραμέτρων, γίνεται επαναληπτικά, βήμα προς βήμα με στόχο το δίκτυο να μπορεί να προσομοιώσει τα δεδομένα που έχει ως είσοδο από τον δάσκαλο. Αν αυτό γίνει εφικτό, τότε μπορούμε να επιτρέψουμε στο δίκτυο να αλληλεπιδράσει με το περιβάλλον χωρίς τη παρουσία του δασκάλου.

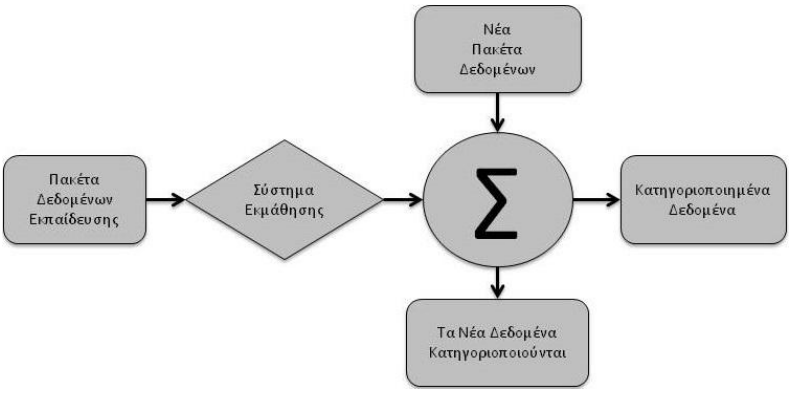
Το σύστημα ή αλλιώς αλγόριθμοι εκπαίδευσης, τροποποιούν τα βάρη με σκοπό την επόμενη φορά που το δίκτυο θα δεχθεί ως είσοδο το ίδιο πρότυπο, η έξοδος που θα παράγει να μοιάζει περισσότερο με τα αναμενόμενα αποτελέσματα εξόδου. Στην Εικόνα 12 μπορούμε να δούμε το πώς συμπεριφέρεται ένα δίκτυο με τη μέθοδο εκπαίδευση με επίβλεψη. Ο ποιο γνωστός αλγόριθμος μάθησης με επίβλεψη είναι ο Back-Propagation [(Michael J.A. Berry, 2004), (Hykin, 1999)] για τον οποίο και θα μιλήσουμε αναλυτικότερα σε επόμενο κεφαλαίο.



Εικόνα 12: Διάγραμμα Ροής Εκπαίδευσης με Επίβλεψη

1.2.4.2 Μάθηση χωρίς επίβλεψη (Unsupervised Learning)

Στη μάθηση χωρίς επίβλεψη, τα νευρωνικά δίκτυα εκπαιδεύονται χωρίς τη παρέμβαση κάποιου δασκάλου. Αυτό σημαίνει ότι από μόνο του το δίκτυο προσαρμόζεται συνεχώς στις νέες εισόδους για να βρει κάποιο μοτίβο ή κάποια συσχέτιση μεταξύ των δεδομένων εισόδου. Με αυτό το τρόπο το δίκτυο δεν εκπαιδεύεται να αντιδρά σε συγκεκριμένα αποτελέσματα με συγκεκριμένο τρόπο, αλλά αποτελεί ουσιαστικά μια αυτο-οργανωτική διαδικασία εκμάθησης. Μόλις το δίκτυο εξοικειωθεί με τις στατιστικές ιδιότητες των δεδομένων εισόδου, αναπτύσσει την ικανότητα να σχηματίζει εσωτερικές αναπαραστάσεις για τη κωδικοποίηση των χαρακτηριστικών των στοιχείων εισόδου και επομένως την ικανότητα να δημιουργεί αυτόματα νέες τάξης (Becker, 1991). Στην Εικόνα 13 μπορούμε να δούμε το πώς συμπεριφέρεται ένα δίκτυο με τη μέθοδο, εκπαίδευση χωρίς επίβλεψη.



Εικόνα 13: Διάγραμμα Ροής Εκπαίδευσης Χωρίς Επίβλεψη

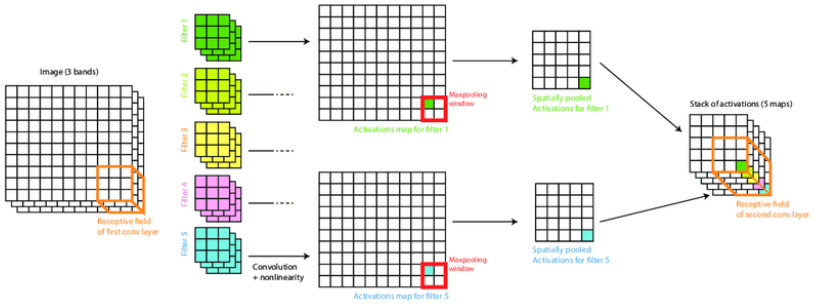
1.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks (CNN))

Ένα CNN αποτελείται από ένα ή περισσότερα επίπεδα συνέλιξης (*convolutional layers*) συχνά μαζί με ένα επίπεδο υποδειγματοληψίας (πχ *pooling*[4]*)* ακολουθούμενο από ένα ή περισσότερα πλήρως συνδεδεμένα επίπεδα όπως συμβαίνει και σε ένα κλασικό πολύ-επίπεδο νευρωνικό δίκτυο. Η αρχιτεκτονική των CNN σχεδιάζεται έτσι ώστε να εκμεταλλεύεται την 2D δομή των δεδομένων εισόδου όπως εικόνες ή άλλα 2D σήματα όπως σήματα ήχου. Αυτό επιτυγχάνεται με τοπικές συνδέσεις και κατάλληλα βάρη ακολουθούμενα από *pooling* προκειμένου να δημιουργηθούν χαρακτηριστικά ανεξάρτητα

[4] *Η σημασία των επιπέδων pooling (συγκέντρωση) θα εξηγηθεί σε επόμενο κεφάλαιο.*

μετατοπίσεων (*translation invariant*). Άλλο ένα προσόν των CNNs είναι ότι εκπαιδεύονται ευκολότερα, σε σύγκριση με άλλα δίκτυα, και αποτελούνται από λιγότερες παραμέτρους σε σύγκριση με αυτές των πλήρως συνδεδεμένων νευρωνικών δικτύων με τον ίδιο αριθμό κρυφών επιπέδων, πράγμα που σημαίνει γρηγορότερη απόδοση του δικτύου.

Θεωρώντας ότι η είσοδος σε ένα επίπεδο συνέλιξης είναι μια εικόνα διαστάσεων *h x w x r* όπου το *h (height)* είναι το ύψος, το *w (wide)* είναι το πλάτος της εικόνας, ενώ το *r* δείχνει τον αριθμό των καναλιών της εικόνας (πχ για εικόνα RGB: r=3). Το επίπεδο συνέλιξης έχει *k* πυρήνες (*kernels*) μεγέθους *n x n x q* όπου το *n* είναι μικρότερο από τις διαστάσεις της εικόνας και το *q* μπορεί να είναι ίδιου μεγέθους με τα κανάλια *r* της εικόνας ή μικρότερου και μπορεί να ποικίλει για κάθε πυρήνα. Το μέγεθος των πυρήνων προκαλεί τοπικά συνδεδεμένη δομή όπου καθένα συνελίσσεται με κάθε εικόνα για να παράγουν *k* χάρτες χαρακτηριστικών (*feature maps*) μεγέθους m–n+1. Κάθε χαρακτηριστικό υποδειγματοληπτείται τυπικά με *mean* ή *max pooling* σε *p x p* συνεχείς περιοχές όπου το *p* παίρνει συνίθως τιμές από 2 μέχρι και 5 για πολύ μεγάλες εικόνες εισόδου. Στην Εικόνα 14 αναπαριστάται ένα πλήρες επίπεδο CNN αποτελούμενο από επίπεδα συνέλιξης και pooling. Οι νευρώνες με το ίδιο χρώμα έχουν παρόμοια βάρη. Αυτό το επίπεδο μαθαίνει με Nf = 5 πυρήνες, μεγέθους 3 × 3 × 3 και εφαρμόζει μια χωρική συγκέντρωση μειώνοντας κατά το ήμισυ τον χάρτη ενεργοποίησης (για λόγους σαφήνειας φαίνονται μόνο οι 2 από τους 5 χάρτες). Στους χάρτες ενεργοποίησης, τα έγχρωμα εικονοστοιχεία (*pixels*) (με πράσινο ή μπλε χρώμα) αντιστοιχούν σε εκείνα που λαμβάνουν πληροφορίες από το πεδίο ευαισθητοποίησης (*receptive field*) που σημειώνονται με πορτοκαλί χρώμα στην εικόνα εισόδου (αριστερά) ((Wu, 2018), (Le Lu, 2017), (Ragav Venkatesan, 2017)).



Εικόνα 14: Σχηματική αναπαράσταση του πρώτου συνελικτικού επιπέδου ενός CNN.

Η λειτουργία των CNN συνοψίζεται στα εξής τέσσερα βήματα:

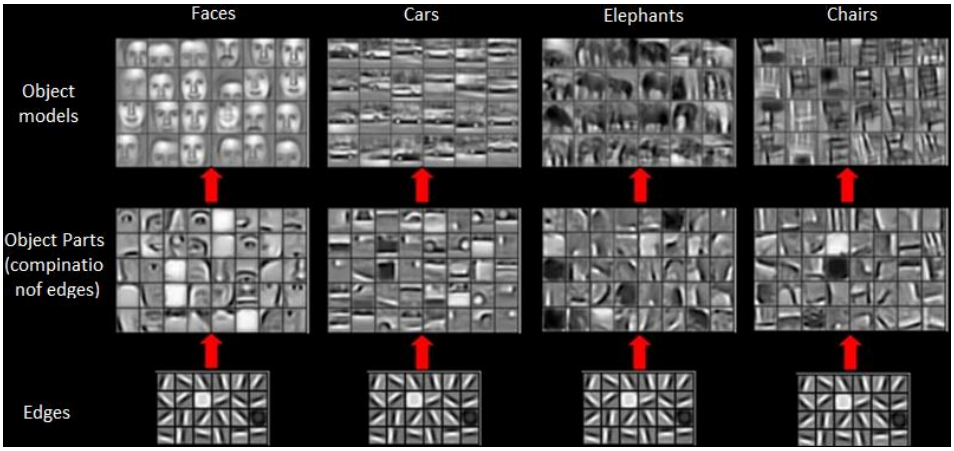
1. Συνέλιξη της εισόδου με φίλτρα που έχουν προκύψει απ' τη διαδικασία Εκμάθησης
2. Εφαρμογή μη γραμμικότητας
3. Χωρικό Pooling
4. Κανονικοποίηση (Normalization)

Η διαδικασία της εκμάθησης των φίλτρων, γίνεται μέσω του αλγορίθμου backpropagation ο οποίος θα περιγραφεί σε επόμενο κεφάλαιο.

1.3.1 Συνελικτικό Επίπεδο (Convolution Layer)

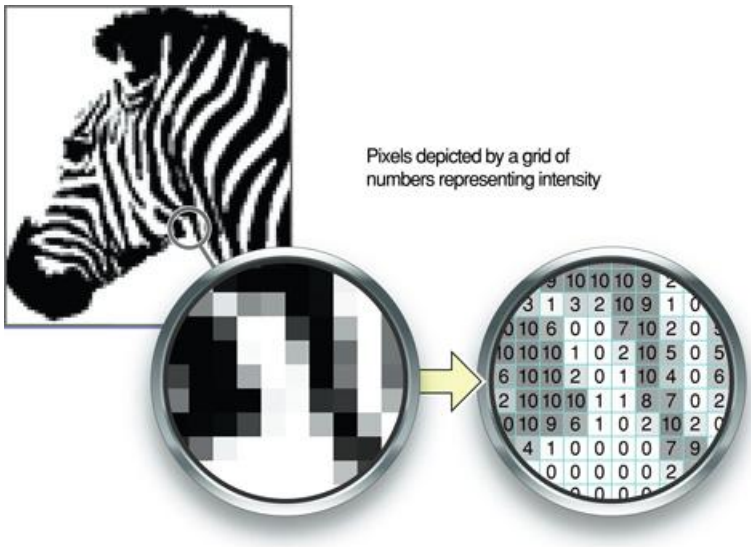
Η κεντρική ιδέα στην οποία βασίζονται τα CNN, είναι ότι τα χαρακτηριστικά πλαισίου (*patch*) της εικόνας, είτε αυτά αφορούν κλίσεις, είτε ακμές (*edges*) κλπ, είναι τοπικά εντοπισμένα. Μερικά από τα πλεονεκτήματα των CNN, είναι η ανεξαρτησία τους ως προς τις μετατοπίσεις της εικόνας σε διάφορες διευθύνσεις (*Translation Invariance*), ο μικρός αριθμός παραμέτρων (βάρη φίλτρων) ή ακόμη και το μεγάλο βήμα μεταξύ των τμημάτων που χρησιμοποιούνται κατά το pooling, το οποίο έχει ως αποτέλεσμα γρηγορότερη εκτέλεση αλγορίθμου και μικρότερες απαιτήσεις μνήμης του συστήματος.

Στην Εικόνα 15, βλέπουμε πως δουλεύει ένα μοντέλο Νευρωνικού Δικτύου το οποίο δέχεται ως είσοδο μια εικόνα. Το μοντέλο ανακαλύπτει προσανατολισμένες ακμές στο 1^{ο} επίπεδο, τμήματα αντικειμένων στο 2^{ο} επίπεδο και πρότυπα ολόκληρων μοντέλων στο 3^{ο} επίπεδο.



Εικόνα 15: Μοντέλο Νευρωνικού Δικτύου για εικόνες

Γνωρίζουμε ότι οι φυσικές εικόνες έχουν την ιδιότητα της *στατικότητας*, δηλαδή η στατιστική που διέπει ένα τμήμα της εικόνας είναι ίδια σε κάθε άλλο μέρος της. Επομένως τα χαρακτηριστικά που μαθαίνουμε για κάποιο τμήμα της εικόνας μπορούν να εφαρμοστούν σε κάθε άλλο μέρος της και μπορούμε να χρησιμοποιήσουμε τα ίδια χαρακτηριστικά για όλα τα τμήματά της (Εικόνα 16).



Εικόνα 16: Εικόνα ως πλέγμα αριθμών

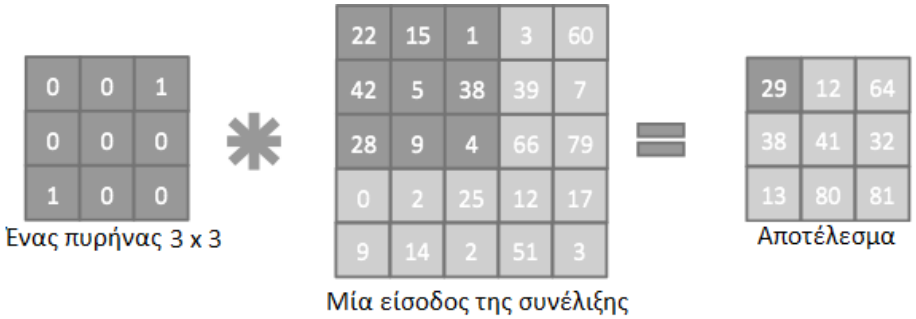
Για να δώσουμε ένα παράδειγμα, ας υποθέσουμε ότι έχουμε ως είσοδο μια εικόνα 5 x 5 και έναν πυρήνα συνέλιξης 3 x 3 όπως φαίνεται στην Εικόνα 17. Θεωρούμε ότι η εικόνα αποτελεί ένα πλέγμα αριθμών (Εικόνα 16). Αν επικαλύψουμε με τον πυρήνα συνέλιξης την εικόνα εισόδου, μπορούμε να υπολογίσουμε το αποτέλεσμα μεταξύ των αριθμών στην ίδια θέση του πυρήνα και της εικόνας εισόδου, και στη συνέχεια αθροίζοντας αυτά τα αποτελέσματα θα πάρουμε το τελικό αποτέλεσμα. Για παράδειγμα, αν επικαλύψουμε με τον πυρήνα την επάνω αριστερή γωνία της εισόδου, το αποτέλεσμα της συνέλιξης σε αυτή τη χωρική τοποθεσία θα είναι:

$$0 \times 22 + 0 \times 15 + 1 \times 1 + 42 \times 0 + 5 \times 0 + 0 \times 38 + 1 \times 28 + 0 \times 9 + 0 \times 4 = 29.$$

Μετακινώντας τον πυρήνα προς τα κάτω, κατά ένα pixel το αποτέλεσμα της συνέλιξης είναι:

$$0 \times 42 + 0 \times 5 + 1 \times 38 + 0 \times 28 + 0 \times 9 + 0 \times 4 + 1 \times 0 + 0 \times 2 + 0 \times 25 = 38.$$

Συνεχίζουμε να μετακινούμε τον πυρήνα προς τα κάτω μέχρι να φτάσει το κάτω όριο του πίνακα εισόδου (*matrix*). Στη συνέχεια ο πυρήνας επιστρέφει στην αρχική του θέση και μετακινείται προς τα δεξιά κατά ένα στοιχείο (*pixel*). Επαναλαμβάνουμε τη διαδικασία για κάθε πιθανή θέση μέχρι να έχουμε μετακινήσει τον πυρήνα στην κάτω δεξιά γωνία της εικόνας. Το ίδιο συμβαίνει και στην Εικόνα 18.



Εικόνα 17: 1ο Παράδειγμα Συνελικτικής Διαδικασίας

Εικόνα 18: 2ο Παράδειγμα Συνελικτικής Διαδικασίας

Η λειτουργία συνέλιξης πραγματοποιείται με παρόμοιο τρόπο και για νευρώνες μεγαλύτερης τάξης. Γενικά, μπορούμε να πούμε ότι αν έχουμε στο *L* επίπεδο είσοδο με διαστάσεις $H^L \times W^L \times D^L$, πυρήνα συνέλιξης διαστάσεων $H \times W \times D^L$, επικαλύπτοντας με

τον πυρήνα την κορυφή του πίνακα της εισόδου στη χωρική θέση (0, 0, 0), υπολογίζουμε τα αποτελέσματα των αντίστοιχων στοιχείων σε όλα τα κανάλια D^L και αθροίζουμε τα αποτελέσματα HWD^L για να πάρουμε το αποτέλεσμα της συνέλιξης σε αυτή τη χωρική τοποθεσία. Στη συνέχεια, μεταφέρουμε τον πυρήνα από πάνω προς τα κάτω και από αριστερά προς τα δεξιά για να ολοκληρώσουμε τη συνέλιξη.

Σε ένα επίπεδο συνέλιξης, συνήθως χρησιμοποιούνται πυρήνες πολλαπλών συνελίξεων. Υποθέτοντας ότι χρησιμοποιούνται D πυρήνες και κάθε πυρήνας έχει χωρικό εύρος H × W, θεωρούμε τους πυρήνες ως συνάρτηση *f*. Η f ορίζεται στο $\mathbb{R}^{\mathrm{H}\times\mathrm{W}\times\mathrm{D}^{\mathrm{L}}\times\mathrm{D}}$. Ομοίως, χρησιμοποιούμε μεταβλητούς δείκτες $0 \le i < H$, $0 \le j < W$, $0 \le d^L < D^L$ και $0 \le d < D$ για να υποδείξουμε ένα συγκεκριμένο στοιχείο των πυρήνων.

Όπως βλέπουμε στην Εικόνα 17, ο πίνακας που παίρνουμε ως έξοδο είναι μικρότερος από τον πίνακα εισόδου, αυτό ισχύει για εισόδους μεγαλύτερες του 1 x 1. Μερικές φορές απαιτείται η έξοδος να έχει το ίδιο μέγεθος με την είσοδο. Για να είναι δυνατό αυτό σε μια είσοδο $H^L \times W^L \times D^L$ και με μέγεθος πυρήνα $\mathrm{H}\times\mathrm{W}\times\mathrm{D}^{\mathrm{L}}\times\mathrm{D}$, το αποτέλεσμα της συνέλιξης θα έχει μέγεθος $(H^L - H + 1) \times (W^L - W + 1) \times D$.

Σε μαθηματικούς όρους, η συνελικτική διαδικασία εκφράζεται με τη παρακάτω εξίσωση:

$$y_{i^{L+1},j_{L+1},d} = \sum_{i=0}^{H-1}\sum_{j=0}^{W-1}\sum_{D^L=0}^{D^L-1} f_{i,j,d^L,d} \times x^L_{i^{L+1}+i,j^{L+1}+j,d^L}$$

Εξίσωση 9: Συνελικτική διαδικασία

Η Εξίσωση 8 επαναλαμβάνεται για όλες τις τιμές $0 \le d < D = D^{L+1}$, και για οποιαδήποτε χωρική θέση (i^{L+1}, j^{L+1}) η οποία ικανοποιεί τα παρακάτω:

$$0 \le i^{L+1} < H^L - H + 1 = H^{L+1}$$

και $$0 \le j^{L+1} < W^L - W + 1 = W^{L+1}.$$

1.3.2 Γιατί χρησιμοποιούμε συνέλιξη

Στην Εικόνα 19 βλέπουμε μια έγχρωμη εικόνα (19α) και τα αποτελέσματα της συνέλιξης χρησιμοποιώντας δύο διαφορετικούς πυρήνες (19β και 19γ). Για τη συνέλιξη χρησιμοποιείται ο παρακάτω συνελικτικός πίνακας 3 x 3:

$$K = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Ο συνελικτικός πυρήνας θα πρέπει να είναι της μορφής 3 x 3 x 3. Όταν εντοπισθεί κάποια οριζόντια ακμή στο σημείο (*x,y*) (π.χ. όταν τα pixel στην θέση (*x+1, y*) και (*x-1, y*) διαφέρουν κατά ένα μεγάλο ποσοστό), αναμένουμε το αποτέλεσμα της συνέλιξης να έχει μεγάλη ένταση. Όπως βλέπουμε στην Εικόνα 18β, το αποτέλεσμα της συνέλιξης όντως επισημαίνει τις οριζόντιες ακμές. Όταν θέσουμε όλα τα κανάλια του συνελικτικού πυρήνα σε K^T , το συνελικτικό αποτέλεσμα ενισχύει τις κατακόρυφες ακμές (Εικόνα 19γ). Ο πίνακας (ή φίλτρο) *K* και K^T αποκαλούνται χειριστές Sobel[5].

Εικόνα 19: Η Επίδραση διαφορετικών συνελικτικών πυρήνων σε έγχρωμη εικόνα

Αν προσθέσουμε ένα βάρος για μεροληψία στη λειτουργία συνέλιξης, μπορούμε να κάνουμε το αποτέλεσμα της συνέλιξης θετικό σε οριζόντιες (ή κατακόρυφες) ακμές προς μια συγκεκριμένη κατεύθυνση (π.χ. μια οριζόντια ακμή της οποίας τα εικονοστοιχεία πάνω από αυτή να είναι φωτεινότερα από τα εικονοστοιχεία κάτω από αυτή), και αρνητικές στις υπόλοιπες περιπτώσεις. Εάν το επόμενο επίπεδο είναι ένα επίπεδο ReLU, η έξοδος του

[5] Ο χειριστής Sobel (Sobel operator) πήρε το όνομά του από τον Irwin Sobel, έναν Αμερικανό ερευνητή στην επεξεργασία ψηφιακής εικόνας.

επόμενου επιπέδου ορίζει πολλές λειτουργίες ανίχνευσης ακμών, οι οποίες ενεργοποιούνται μόνο σε οριζόντιες ή κάθετες ακμές με ορισμένη κατεύθυνση.

Όταν εμβαθύνουμε περισσότερο στα βαθιά δίκτυα, τα επόμενα επίπεδα μπορούν να μάθουν να ενεργοποιούνται μόνο για συγκεκριμένα (αλλά και περισσότερο σύνθετα) μοτίβα, π.χ. ομάδες ακμών που σχηματίζουν ένα συγκεκριμένο μοτίβο. Αυτό οφείλεται στο ότι οποιοδήποτε στοιχείο στο επίπεδο *L + 1* λαμβάνει υπόψη του τη συνδυασμένη επίδραση πολλών χαρακτηριστικών του επιπέδου *L.* Οι περιπλοκότεροι αυτοί σχεδιασμοί θα συνδυαστούν περαιτέρω από βαθύτερα επίπεδα ώστε να ενεργοποιηθούν για μέρη αντικειμένων μεγαλύτερης σημασιολογίας ή ακόμη και για ένα συγκεκριμένο τύπο αντικειμένων, π.χ. σκύλος, γάτα, αυτοκίνητο, δέντρο, παραλία, κλπ.

Ένα ακόμα πλεονέκτημα του επιπέδου συνέλιξης είναι ότι όλες οι χωρικές τοποθεσίες μοιράζονται τον ίδιο πυρήνα συνέλιξης, το οποίο μειώνει σημαντικά τον αριθμό των παραμέτρων που χρειάζονται για ένα επίπεδο συνέλιξης. Για παράδειγμα, εάν εμφανίζονται πολλά σκυλιά σε μια εικόνα εισόδου, το ίδιο χαρακτηριστικό (π.χ. κεφάλι σκύλου) μπορεί να ενεργοποιηθεί σε πολλαπλές θέσεις, που αντιστοιχούν σε διαφορετικά αλλά όμοια αντικείμενα (κεφάλια διαφορετικών σκύλων).

Σε μια αρχιτεκτονική βαθύ νευρωνικού δικτύου, η συνέλιξη ενθαρρύνει επίσης την κοινή χρήση παραμέτρων. Για παράδειγμα, ας υποθέσουμε ότι το «μοτίβο κεφάλι σκύλου» και το «μοτίβο κεφάλι γάτας» είναι δύο χαρακτηριστικά γνωστά σε ένα βαθύ συνελικτικό δίκτυο. Το CNN χρειάζεται να διαθέσει δύο ομάδες διαχωρισμένων παραμέτρων (π.χ. πυρήνες συνέλιξης σε πολλαπλά επίπεδα) γι 'αυτά. Τα κατώτερα επίπεδα του CNN μπορούν να μάθουν το «μοτίβο μάτι» και «μοτίβο υφή τριχώματος ζώου», και τα δύο μοιράζονται περισσότερο αφηρημένα χαρακτηριστικά. Εν ολίγοις, ο συνδυασμός των πυρήνων συνέλιξης και των βαθιών και ιεραρχημένων δομών είναι πολύ αποτελεσματικός στην εκμάθηση παραστάσεων (χαρακτηριστικών) από εικόνες για εργασίες οπτικής αναγνώρισης ή καλύτερα κατηγοριοποίησης εικόνων.

Αξίζει να σημειωθεί ότι παρόλο που έχουν χρησιμοποιηθεί φράσεις όπως το «μοτίβο κεφάλι σκύλου», η παράσταση ή το χαρακτηριστικό που έχει μάθει ένα CNN μπορεί να μην

αντιστοιχεί ακριβώς στη σημασιολογική έννοια όπως το «κεφάλι σκύλου». Ένα χαρακτηριστικό ενός CNN μπορεί να ενεργοποιείται συχνά για κεφάλια σκύλων και συχνά να απενεργοποιείται για άλλα μοτίβα. Ωστόσο, υπάρχει πάντα η πιθανότητα για λάθος ενεργοποιήσεις σε ορισμένες τοποθεσίες (*false activations*), καθώς και πιθανές μη-ενεργοποιήσεις (*deactivations*) για πραγματικά κεφάλια σκύλων.

Στη πραγματικότητα, μια κύρια ικανότητα των CNN (ή γενικότερα στην βαθιά εκμάθηση) είναι η κατανεμημένη αναπαράσταση *(distributed representation)*. Για παράδειγμα, αν υποθέσουμε ότι στόχος μας είναι να αναγνωρίσουμε *N* διαφορετικούς τύπους αντικειμένων και ένα CNN εξάγει *M* χαρακτηριστικά από οποιαδήποτε εικόνα εισόδου τότε οποιοδήποτε από τα *M* αντικείμενα μπορεί να χρησιμοποιηθεί για την αναγνώριση όλων των *N* αντικειμένων και η αναγνώριση ενός αντικειμένου *N* να απαιτεί τη συνδυασμένη προσπάθεια όλων των *M* χαρακτηριστικών.

1.3.3 Η συνέλιξη ως πίνακας[6]

Η Εξίσωση 8 είναι αρκετά περίπλοκη. Μπορούμε να αναλύσουμε το x^L και να απλοποιήσουμε τη συνέλιξη ως πίνακα αριθμών.

Ας υποθέσουμε την ειδική περίπτωση όπου $D^L = D = 1, H = W = 2, και\ H^L = 3, W^L = 4.$ Δηλαδή, θεωρούμε ότι η συνέλιξη αποτελείται από ένα κανάλι 3 x 4 με ένα φίλτρο 2 x 2. Χρησιμοποιώντας ως παράδειγμα την Εικόνα 17, έχουμε:

$$\begin{bmatrix} 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 1 \\ 7 & 8 & 9 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 12 & 16 & 11 \\ 24 & 28 & 17 \end{bmatrix}$$

Όπου ο πρώτος πίνακας δηλώνεται ως *A* και το σύμβολο * υποδηλώνει το χειριστή της συνέλιξης.

Εάν τώρα τρέξουμε την εντολή **B=im2col(A,[2 2])** στο Matlab, παίρνουμε ως αποτέλεσμα το πίνακα *B* ο οποίος είναι μια εκτεταμένη έκδοση του πίνακα *A:*

[6] Για την επεξήγηση των παραδειγμάτων έγινε η χρήση του Matlab (Έκδοση: R2018b https://uk.mathworks.com/downloads/).

$$B = \begin{bmatrix} 1 & 4 & 2 & 5 & 3 & 6 \\ 4 & 7 & 5 & 8 & 6 & 9 \\ 2 & 5 & 3 & 6 & 1 & 1 \\ 5 & 8 & 6 & 9 & 1 & 1 \end{bmatrix}$$

Είναι εμφανές ότι η πρώτη στήλη του πίνακα *B* αντιστοιχεί στη πρώτη περιοχή 2 x 2 του *A,* και από χωρικής άποψης απαντά στο (i^{L+1}, j^{L+1})=(0,0). Ομοίως, στις περιοχές του *A* με (i^{L+1}, j^{L+1}) είναι (1,0), (0,1), (1,1), (0,2) και (1,2) αντίστοιχα. Δηλαδή, η συνάρτηση **im2col** του Matlab επεκτείνει τα απαιτούμενα στοιχεία διενεργώντας την κάθε ανεξάρτητη συνέλιξη σε κάθε στήλη στο πίνακα *B.* Αξίζει να σημειωθεί ότι η παράμετρος [2 2] ορίζει το μέγεθος του πυρήνα συνέλιξης.

Σε περίπτωση που μετατρέψουμε το πυρήνα συνέλιξης σε διάνυσμα (κατά στήλη) $(1,1,1,1)^T$, έχουμε[7]:

$$B^T \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 12 \\ 24 \\ 16 \\ 28 \\ 11 \\ 17 \end{bmatrix}$$

Το αποτέλεσμα του διανύσματος αυτού, εάν μετασχηματίσουμε τον παραπάνω πίνακα, θα είναι ακριβώς το ίδιο με τον συνελικτικό πίνακα που έχουμε ως έξοδο στην Εικόνα 17.

Δηλαδή, ο χειριστής συνέλιξης είναι στη πραγματικότητα ένας γραμμικός χειριστής. Μπορούμε να πολλαπλασιάσουμε τον διευρυμένο πίνακα εισόδου και το διανυσματικό φίλτρο για να πάρουμε ως αποτέλεσμα ένα διάνυσμα, και διαμορφώνοντας σωστά το διάνυσμα, παίρνουμε το σωστό αποτέλεσμα της συνέλιξης.

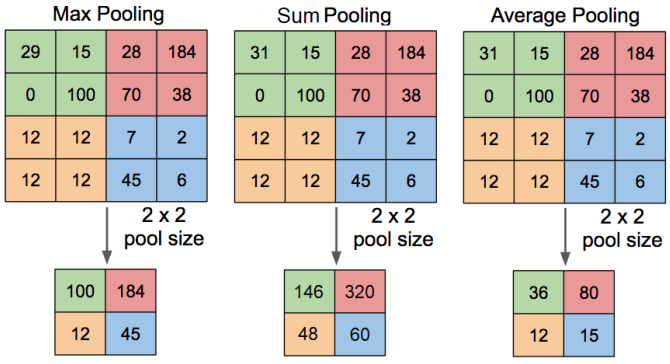
1.3.4 Συγκέντρωση (Pooling)

Όπως αναφέρθηκε και σε προηγούμενο κεφάλαιο, τα επίπεδα συγκέντρωσης (pooling layers) στα CNNs, συνοψίζουν τις εξόδους γειτονικών ομάδων νευρώνων εντός ενός πλαισίου (patch) με μια αντιπροσωπευτική τιμή, ενώ συνήθως τα γειτονικά παράθυρα δεν

[7] Η ανάλυση αυτή βασίστηκε στο εγχειρίδιο χρήσης του MatConvNet (http://arxiv.org/abs/1412.4564, το οποίο είναι βασισμένο στο Matlab). Η μετατροπή im2col είναι ισοδύναμη με την im2row.

επικαλύπτονται. Πρόκειται ουσιαστικά για μια διαδικασία υπο-δειγματοληψίας των δεδομένων ενώ για καλύτερη κατανόηση της διαδικασίας μπορούμε να φανταστούμε ένα επίπεδο pooling σαν ένα πλέγμα pooling νευρώνων τοποθετημένων σε απόσταση *s* pixels, καθένας από τους οποίους συνοψίζει μια περιοχή *z x z* με κέντρο τον ίδιο τον νευρώνα. Θέτοντας *s = z* λαμβάνουμε τα κλασικά αποτελέσματα των κοινώς χρησιμοποιούμενων, μη επικαλυπτόμενων παραθύρων pooling, ενώ για *s < z* λαμβάνουμε τιμές από επικαλυπτόμενα παράθυρα.

Το pooling αποτελεί μια πολύ βασική λειτουργία για κάθε CNN, αφού απλοποιεί πολύ τη διαδικασία εκμάθησης λόγω της σημαντικής μείωσης του όγκου των δεδομένων κι επομένως του αριθμού των απαιτούμενων πράξεων. Οι επικρατέστερες κατηγορίες του pooling είναι το max, sum και average pooling, ενώ μπορεί τα παράθυρα που χρησιμοποιούνται να επικαλύπτονται ή και όχι ανάλογα με τις ανάγκες του προβλήματος. Στην Εικόνα 20 μπορούμε να δούμε τις πιο συχνές μορφές pooling με τη χρήση φίλτρου 2 x 2, με βήμα 2, το οποίο καταφέρνει με αυτό το τρόπο να αποφύγει το 75% των ενεργοποιήσεων. Η διαδικασία του pooling, εκτός από τη μείωση του μεγέθους των δεδομένων, μας δίνει τη δυνατότητα προσθήκης περισσότερης πληροφορίας στην αρχική εικόνα μέσω των αρχικών διαστάσεων ενώ είναι ανεξάρτητη μικρών μετασχηματισμών.



Εικόνα 20: Έφαρμογή max, sum και average pooling (χωρίς επικάλυψη) σε πίνακα

Γενικά, για ένα επίπεδο συνέλιξης ισχύουν τα παρακάτω:

1. Δέχεται ως είσοδο αντικείμενα με μέγεθος $H_1 \times W_1 \times D_1$
2. Για τη λειτουργία του απαιτεί τις εξής παραμέτρους:

α. τη χωρική έκταση *F,*

β. το βήμα *S*

3. Το αποτέλεσμα που εξάγει έχει μέγεθος $H_2 \times W_2 \times D_2$ *όπου*,

α. $W_2 = \frac{W_1 - F}{S} + 1$

β. $H_2 = \frac{H_1 - F}{S} + 1$

γ. $D_2 = D_1$

ΚΕΦΑΛΑΙΟ 2

2. Από την απλή ταξινόμηση στην αναγνώριση εικόνων

2.1 Εισαγωγή

Οι πρόσφατες προσεγγίσεις για την αναγνώριση αντικειμένων κάνουν ουσιαστική χρήση των μεθόδων μηχανικής μάθησης. Για να βελτιώσουμε την απόδοσή τους, μπορούμε να συλλέξουμε μεγαλύτερα σύνολα δεδομένων, να εκπαιδεύσουμε πιο ισχυρά μοντέλα και να χρησιμοποιήσουμε καλύτερες τεχνικές για τη πρόληψη της υπερφόρτωσης. Μέχρι πρόσφατα, τα έτοιμα πακέτα εικόνων ήταν σχετικά μικρά, της τάξης δεκάδων χιλιάδων εικόνων π.χ., NORB (Y. LeCun F. H., 2004), Caltech-101/256 [(L. Fei-Fei, 2007), (G. Griffin, 2007)] και CIFAR-10/100 [(Krizhevsky, 2009),(Krizhevsky,2010)]. Οι απλές εργασίες αναγνώρισης μπορούν να λυθούν αρκετά εύκολα με σύνολα δεδομένων αυτού του μεγέθους. Για παράδειγμα, ο τρέχων ρυθμός σφάλματος στην εντολή αναγνώρισης ψηφίων MNIST (<0,3%) πλησιάζει του ανθρώπου (D. Cire¸san, 2012). Σε πραγματικές συνθήκες όμως τα αντικείμενα παρουσιάζουν μεγάλη ποικιλομορφία, έτσι για να είναι δυνατή η εκπαίδευση μοντέλων για την αναγνώριση αυτών είναι απαραίτητο να χρησιμοποιηθούν πολύ μεγαλύτερα σύνολα δεδομένων. Οι αδυναμίες των δεδομένων μικρών συνόλων εικόνας έχουν αναγνωριστεί (π.χ., (N. Pinto D. C., 2008)), αλλά μόνο πρόσφατα κατέστη δυνατή η κατασκευή πακέτων δεδομένων με ετικέτες εκατομμυρίων

εικόνων. Τα νέα μεγαλύτερα σύνολα δεδομένων περιλαμβάνουν το LabelMe (B.C. Russell, 2008), το οποίο αποτελείται από εκατοντάδες χιλιάδες εικόνες και το ImageNet (J. Deng W. D.-F., 2009), το οποίο αποτελείται από περισσότερες από 15 εκατομμύρια εικόνες υψηλής ανάλυσης σε πάνω από 22.000 διαφορετικές κατηγορίες.

Για να μάθουμε χιλιάδες αντικείμενα από εκατομμύρια εικόνες, χρειαζόμαστε ένα μοντέλο με μεγάλη ικανότητα εκμάθησης. Ωστόσο, η πολυπλοκότητα της αναγνώρισης αντικειμένων σημαίνει ότι αυτό το πρόβλημα δεν μπορεί να προσδιοριστεί ακόμη και από ένα σύνολο δεδομένων τόσο μεγάλο όσο το ImageNet. Τα CNNs αποτελούν μια κατηγορία μοντέλων η οποία εκμεταλλεύεται είδη υπάρχοντα μοντέλα [(K. Jarrett, 2009), (Krizhevsky, 2009), (Y. LeCun F. H., 2004), (H. Lee, 2009), (Y. Le Cun, 1990), (N. Pinto D. D., 2009), (S.C. Turaga, 2010)]. Η χωρητικότητά τους μπορεί να ελεγχθεί μεταβάλλοντας το βάθος και το εύρος τους, ενώ παράλληλα πραγματοποιούν ισχυρές και ως επί το πλείστον ορθές υποθέσεις σχετικά με τη φύση των εικόνων (δηλαδή τη στασιμότητα των στατιστικών στοιχείων και την αλληλεξάρτηση των εικονοστοιχείων). Έτσι, συγκρίνοντας τα τυποποιημένα feed forward νευρωνικά δίκτυα με παρόμοια μεγέθη επιπέδων με τα CNN, παρατηρούμε ότι τα CNN έχουν πολύ λιγότερες συνδέσεις και παραμέτρους και έτσι είναι ευκολότερο να εκπαιδευτούν, ενώ η θεωρητικά καλύτερη απόδοση τους πιθανόν να είναι ελαφρώς χειρότερη.

Παρά τις ελκυστικές ιδιότητες των CNN και παρά τη σχετική απόδοση της αρχιτεκτονικής τους, εξακολουθούν να είναι απαγορευτικά για εφαρμογή σε μεγάλη κλίμακα με εικόνες υψηλής ανάλυσης. Ευτυχώς, οι τρέχουσες GPU, σε συνδυασμό με μια εξαιρετικά βελτιστοποιημένη υλοποίηση της συνέλιξης 2D, αποτελούν έναν αρκετά ισχυρό συνδυασμό για να διευκολυνθεί η εκπαίδευση ιδιαίτερα μεγάλων CNN. Πρόσφατα σύνολα δεδομένων όπως το ImageNet και το AlexNet (Krizhevsky A. S., 2012) περιέχουν επαρκή ταξινομημένα παραδείγματα για την εκπαίδευση τέτοιων μοντέλων.

2.2 ImageNet

Το ImageNet είναι ένα σύνολο δεδομένων με πάνω από 15 εκατομμύρια ταξινομημένες εικόνες υψηλής ανάλυσης που ανήκουν σε περίπου 22.000 κατηγορίες. Οι εικόνες

συλλέχθηκαν από τον ιστό και επισημάνθηκαν από ανθρώπους που χρησιμοποίησαν το εργαλείο Amazon's Mechanical Turk-crowd-sourcing (J. Shotton, 2006). Το ILSVRC χρησιμοποιεί ένα υποσύνολο του ImageNet με περίπου 1000 εικόνες για κάθε μία από τις 1000 κατηγορίες. Συνολικά, υπάρχουν περίπου 1,2 εκατομμύρια εικόνες εκπαίδευσης, 50.000 εικόνες επικύρωσης και 150.000 εικόνες δοκιμών.

2.3 AlexNet

Η αρχιτεκτονική του δικτύου AlexNet συνοψίζεται στο σχήμα 2. Περιέχει οκτώ μαθησιακά επίπεδα - πέντε συνελικτικές και τρεις πλήρως συνδεδεμένες. Παρακάτω, περιγράφουμε μερικά από τα μυθιστορήματα ή τα ασυνήθιστα χαρακτηριστικά της αρχιτεκτονικής του δικτύου μας.

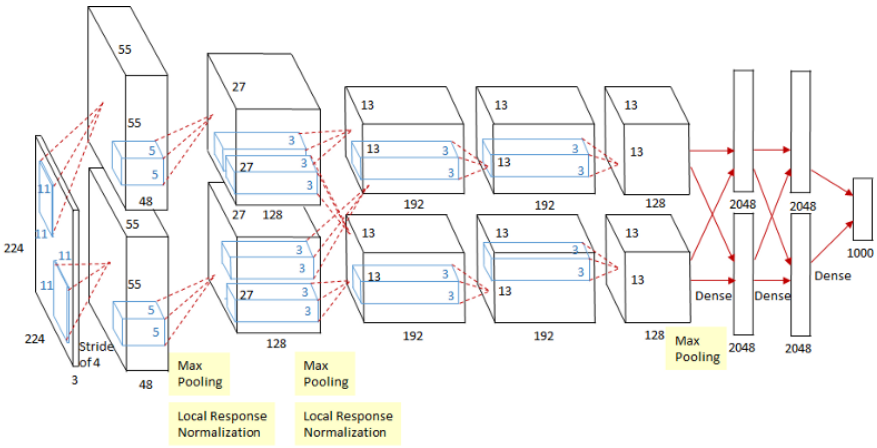
Το 2012, προτάθηκε ένα βαθύτερο και ευρύτερο μοντέλο CNN σε σύγκριση με το LeNet και κέρδισε τη δυσκολότερη πρόκληση του ImageNet, την αναγνώριση οπτικών αντικειμένων που ονομάζεται ILSVRC (Krizhevsky A. S., 2012). Το AlexNet κατάφερε να ξεχωρίσει ανάμεσα από όλες τις παραδοσιακές μηχανές μάθησης και μηχανικής όρασης. Αποδείχτηκε μια τεράστια καινοτομία στο τομέα της εκμάθησης μηχανών και της μηχανικής όρασης για την οπτική αναγνώριση και της διαδικασίας ταξινόμησης καθώς επίσης αποτελεί και ένα σημείο της ιστορίας από το οποίο το ενδιαφέρον για τη βαθιά μάθηση αυξήθηκε κατακόρυφα.

2.3.1 Η Αρχιτεκτονική του AlexNet

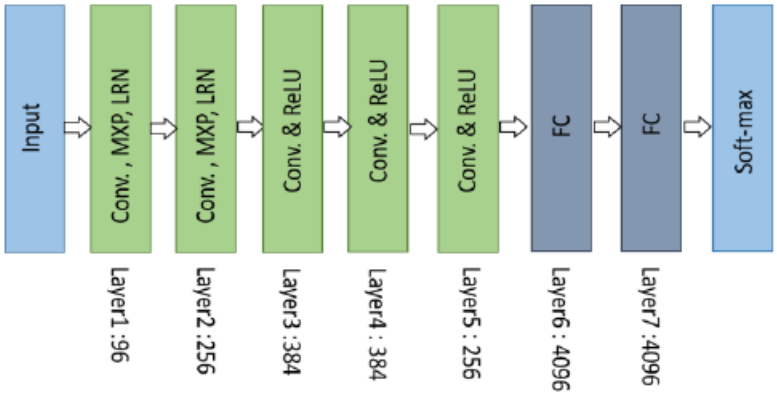
Μία γενική αρχιτεκτονική του AlexNet φαίνεται στην Εικόνα 21 και αναλυτικότερα στην Εικόνα 20α. Το πρώτο επίπεδο συνέλιξης φιλτράρει την εικόνα εισόδου 224 × 224 × 3 με 96 πυρήνες μεγέθους 11 × 11 × 3 με ένα βήμα της τάξης των 4 pixels (αυτή είναι η απόσταση μεταξύ των κέντρων των γειτονικών νευρώνων σε ένα χάρτη πυρήνα). Το δεύτερο συνελικτικό επίπεδο παίρνει ως είσοδο (έχοντας κοινωνικοποιηθεί και συγκεντρωθεί (*pooling*)) την έξοδο του πρώτου συνελικτικού επιπέδου και το φιλτράρει με 256 πυρήνες μεγέθους 5 × 5 × 48. Το τρίτο, τέταρτο και πέμπτο συνελικτικό επίπεδα συνδέονται μεταξύ τους χωρίς παρεμβάσεις συγκέντρωσης ή εξομάλυνσης. Το τρίτο συνελικτικό επίπεδο έχει

384 πυρήνες μεγέθους 3x3x256 που συνδέονται με τις (κανονικοποιημένες, συγκεντρωμένες) εξόδους του δεύτερου επιπέδου. Το τέταρτο επίπεδο έχει 384 πυρήνες μεγέθους 3 × 3 × 192 και το πέμπτο επίπεδο έχει 256 πυρήνες μεγέθους 3 × 3 × 192. Τα πλήρως συνδεδεμένα επίπεδα έχουν 4096 νευρώνες το καθένα (Εικόνα 21α, 21).

Δύο νέες έννοιες, η LRN και η εγκατάλειψη (dropout), εισάγονται σε αυτό το δίκτυο. Το LRN μπορεί να εφαρμοστεί με δύο διαφορετικούς τρόπους. Πρώτα εφαρμογή σε χάρτες μονών καναλιών ή χαρτών χαρακτηριστικών, όπου επιλέγεται ένα πλαίσιο N × N από τον ίδιο χάρτη χαρακτηριστικών και κανονικοποιείται με βάση τις γειτονικές τιμές. Δεύτερον, το LRN μπορεί να εφαρμοστεί μεταξύ των καναλιών ή των χαρτών χαρακτηριστικών (γειτνίαση κατά μήκος της τρίτης διάστασης, αλλά με ένα μόνο pixel).



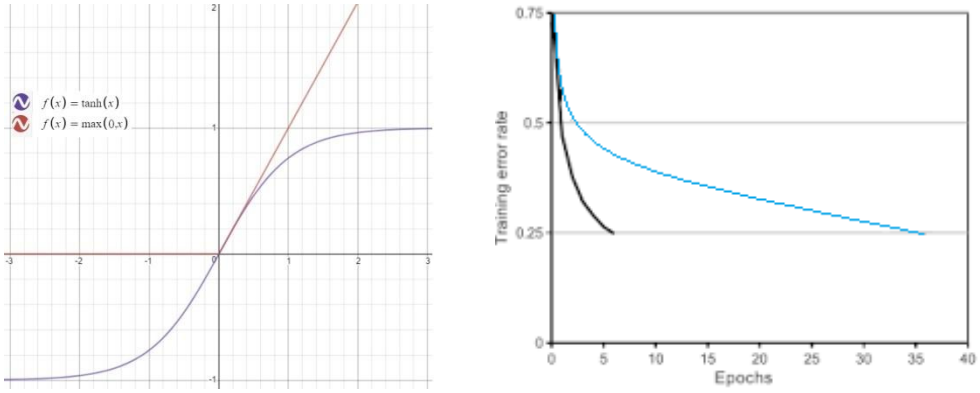
Εικόνα 21α



Εικόνα 21: Η αρχιτεκτονική του AlexNet

2.3.2 ReLU μη Γραμμικότητα

Ο συνήθης τρόπος για να μοντελοποιήσουμε την έξοδο ενός νευρώνα ως μια συνάρτηση της εισόδου *x* είναι με *f(x) = tanh(x)* ή $f(x)=(1+e^{-x})^{-1}$ (Εικόνα 21 (Αριστερά)). Όσον αφορά το χρόνο εκπαίδευσης, αυτές οι μη γραμμικές είναι πολύ αργές από τη μη γραμμικότητα *f(x) = max(0,x)*. Όπως είχαν αναφέρει οι Nair και Hinton (Hinton V. N., 2010), οι νευρώνες που εμφανίζουν αυτήν τη μη γραμμικότητα ονομάζονται Διορθωμένες Γραμμικές Μονάδες (Rectified Linear Units: ReLUs). Τα βαθιά συνελικτικά Νευρωνικά Δίκτυα με ReLUs εκπαιδεύονται αρκετές φορές ταχύτερα σε σχέση με ισοδύναμα δίκτυα με μονάδες *tanh*. Αυτό φαίνεται στην Εικόνα 22, η οποία δείχνε τον αριθμό των επαναλήψεων (*iterations*) που απαιτούνται για να φτάσει το σφάλμα εκπαίδευσης σε ποσοστό 25% για ένα δίκτυο.



Εικόνα 22: (Αριστερά) Οι συναρτήσεις *f(x)*. (Δεξιά) Ένα Δίκτυο Τεσσάρων Επιπέδων με ReLUs (μαύρο χρώμα) φτάνει το 25% του σφάλματος εκπαίδευσης έξι φορές ταχύτερα σε σχέση με ένα δίκτυο *tanh (μπλε χρώμα)*.

2.3.3 Κανονικοποίηση Τοπικής Απόκρισης

Τα ReLUs έχουν την ιδιαιτερότητα να μη χρειάζονται κανονικοποίηση κατά την είσοδο των δεδομένων και έτσι αποφεύγουν να υπερφορτωθούν. Στη περίπτωση που ορισμένα παραδείγματα εκπαίδευσης παράγουν μια θετική είσοδο σε ένα ReLU, αυτός ο νευρώνας θα εκπαιδευτεί. Θεωρώντας ως $a_{x,y}^{i}$ τη δραστηριότητα ενός νευρώνα εφαρμόζοντας τον πυρήνα *i* στη θέση (x, y) και στη συνέχεια εφαρμόζοντας τη μη γραμμικότητα του ReLU, η κανονικοποιημένη απόκριση $b_{x,y}^{i}$ δίνεται από τη παρακάτω εξίσωση:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + a \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a_{x,y}^{j})^2\right)^{\beta}}$$

Εξίσωση 10: Κανονικοποιημένη Απόκριση

όπου το άθροισμα τρέχει πάνω από n γειτονικούς πυρήνες στην ίδια χωρική θέση, και *N* είναι το σύνολο των πυρήνων στο επίπεδο. Η ταξινόμηση των χαρτών πυρήνα είναι φυσικά αυθαίρετη και αποσαφηνίζεται πριν ξεκινήσει η εκπαίδευση. Αυτός ο τύπος κανονικοποίησης εφαρμόζει μια μορφή πλευρικής αναστολής εμπνευσμένη από τους φυσικούς νευρώνες, δημιουργώντας ανταγωνισμό για μεγάλες δραστηριότητες μεταξύ των νευρώνων εξόδου. Οι σταθερές *k, n, α* και *β* είναι υπερ-παράμετροι των οποίων οι τιμές προσδιορίζονται με τη χρήση ενός συνόλου επικύρωσης.

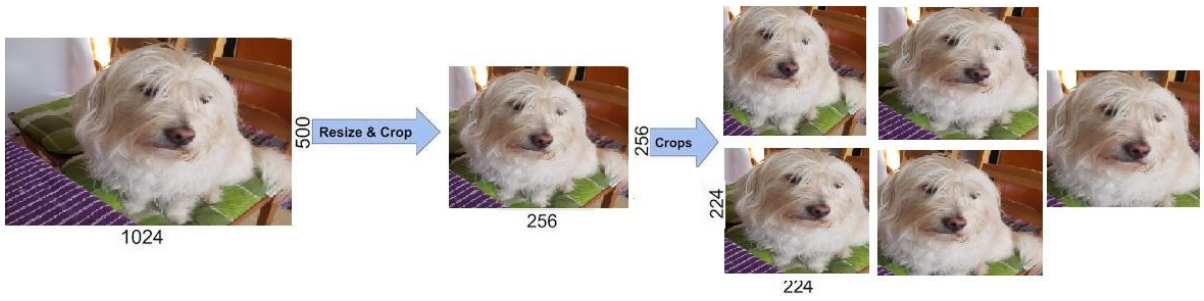
2.3.4 Υπερφόρτωση

Το Alexnet διαθέτει περίπου 60 εκατομμύρια παραμέτρους και 650.000 νευρώνες. Για να αποφύγουν την υπερφόρτωση του δικτύου εφαρμόζονται δύο κύριες μέθοδοι οι οποίες επιτρέπουν τις μετασχηματισμένες εικόνες να παραχθούν από τις αρχικές με πολύ λίγους υπολογισμούς. Με αυτό το τρόπο οι μετασχηματισμένες εικόνες δεν απαιτείται να αποθηκεύονται στον δίσκο. Συγκεκριμένα, οι μετασχηματισμένες εικόνες δημιουργούνται μέσα από κώδικα της Python στην CPU ενώ η GPU αναλαμβάνει την εκπαίδευση του δικτύου με τη προηγούμενη παρτίδα εικόνων. Έτσι, αυτά τα συστήματα αύξησης των δεδομένων, στη πραγματικότητα δεν απαιτούν υπολογιστική ισχύ.

Η πρώτη μορφή αύξησης δεδομένων απαρτίζεται από την δημιουργία μεταφράσεων εικόνας (*image translations*) και οριζόντιων ανακλάσεων. Αυτό επιτυγχάνεται με την εξαγωγή τυχαίων 224 × 224 patches (και των οριζόντιων ανακλάσεών τους) από τις εικόνες διαστάσεων 256 × 256 και εκπαιδεύοντας το δίκτυο με αυτά τα εξαγόμενα κομμάτια[8]. Αυτό αυξάνει το μέγεθος του πακέτου εκπαίδευσης με συντελεστή 2048[9], αν και τα προκύπτοντα παραδείγματα εκπαίδευσης συνεχίζουν να είναι αλληλεξαρτώμενα. Κατά την

[8] Αυτός είναι και ο λόγος για τον οποίο οι εικόνες εισόδου (Εικόνα 20) έχουν διαστάσεις 224 × 224 × 3.
[9] Με εικόνα: (256-224) ² = 32² = 1024 και με οριζόντια ανάκλαση: 1024 × 2 = 2048.

δοκιμή, το δίκτυο κάνει μια πρόβλεψη με την εξαγωγή πέντε πλαισίων μεγέθους 224 × 224 (τέσσερα γωνιακά πλαίσια και ένα κεντρικό) καθώς επίσης και τις οριζόντιες αντανακλάσεις (δέκα πλαίσια στο σύνολο) και υπολογίζει το μέσο όρο των προβλέψεων που πραγματοποιήθηκαν (Εικόνα 23).



Εικόνα 23: Ενδεικτική διαδικασία εκπαίδευσης. Το κάθε νέο πλαίσιο που δημιουργείται (Δεξιά) διαφέρει από τα άλλα

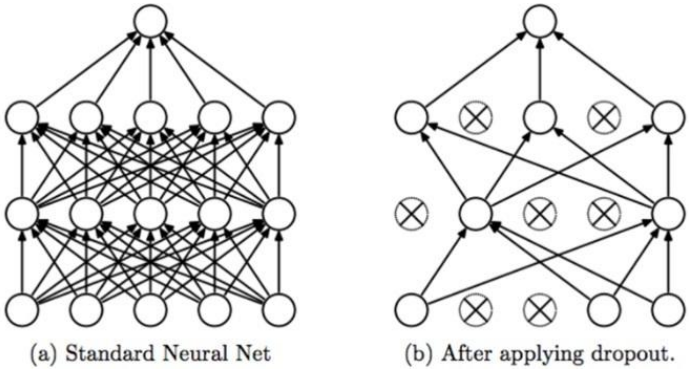
Η δεύτερη μορφή αύξησης δεδομένων συνίσταται στην αλλαγή της έντασης των καναλιών RGB στις εικόνες εκπαίδευσης. Συγκεκριμένα, εκτελείται η ανάλυση PCA (*Principal Components Analysis*) (Richardson, 2009) στο σύνολο των τιμών των εικονοστοιχείων RGB καθ 'όλη τη διάρκεια του εκπαίδευσης με το ImageNet. Σε κάθε εικόνα εκπαίδευσης, προστίθενται πολλαπλάσια τα κύρια συστατικά που βρέθηκαν, με βάρη ανάλογα με τις αντίστοιχες τιμές, από τις οποίες προέρχεται μια τυχαία μεταβλητή Gaussian με μέσο όρο 0 και τυπική απόκλιση 0,1. Επομένως, σε κάθε RGB εικονοστοιχείο της εικόνας $I_{x,y} = [I_{xy}^R, I_{xy}^G, I_{xy}^B]$ προστίθεται:

$$[p_1, p_2, p_3]\,[\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

όπου τα p_i και λ_i αποτελούν τα *i* ιδιοδιανύσματα και ιδιοτιμές του 3 × 3 πίνακα συνάφειας των τιμών του εικονοστοιχείου RGB, αντίστοιχα, το α_i είναι η προαναφερθείσα τυχαία μεταβλητή. Κάθε α_i υπολογίζεται μόνο μία φορά για όλα τα εικονοστοιχεία μιας συγκεκριμένης εικόνας εκπαίδευσης έως ότου η εικόνα χρησιμοποιηθεί ξανά για εκπαίδευση, όπου και υπολογίζεται εκ νέου. Αυτή η μέθοδος στηρίζεται στο γεγονός ότι η ταυτότητα των αντικειμένων, στις φυσικές εικόνες, παραμένει αμετάβλητη στις αλλαγές της έντασης και του χρώματος του φωτισμού.

2.3.5 Εγκατάλειψη Δεδομένων (Dropout)

Ο συνδυασμός των προβλέψεων πολλών διαφορετικών μοντέλων είναι ένας πολύ επιτυχημένος τρόπος για τη μείωση των σφαλμάτων κατά τον έλεγχο [(Koren, 2007), (A. Berg, 2010)], αλλά φαίνεται να είναι πολύ απαιτητικό για μεγάλα νευρωνικά δίκτυα που ήδη χρειάζονται αρκετές ημέρες για να εκπαιδευτούν. Ωστόσο υπάρχει μια νέα τεχνική η οποία ονομάζεται Εγκατάλειψη (G.E. Hinton, 2012), η οποία μηδενίζει της εξόδους του κάθε κρυμμένου νευρώνα με πιθανότητα μικρότερη του 0,5. Οι νευρώνες που απορρίφθηκαν με αυτό το τρόπο δεν προωθούν δεδομένα σε επόμενο επίπεδο και δεν συμμετέχουν στην ανατροφοδότηση. Έτσι, κάθε φορά που παρουσιάζεται μια είσοδος, το νευρωνικό δίκτυο δοκιμάζει μια διαφορετική αρχιτεκτονική, αλλά όλες αυτές οι αρχιτεκτονικές μοιράζονται τα ίδια βάρη. Αυτή η τεχνική μειώνει τις πολύπλοκες συν-προσαρμογές των νευρώνων, καθώς ένας νευρώνας δεν μπορεί να βασιστεί στη παρουσία άλλων συγκεκριμένων νευρώνων. Οι νευρώνες επομένως, είναι αναγκασμένοι να μάθουν μια πληθώρα χαρακτηριστικών που είναι χρήσιμα σε συνδυασμό με πολλά διαφορετικά τυχαία υποσύνολα των άλλων νευρώνων. Κατά την διάρκεια της δοκιμής, χρησιμοποιούνται όλοι οι νευρώνες αλλά πολλαπλασιάζονται οι αποδόσεις τους κατά 0.5, το οποίο αποτελεί μια εύλογη προσέγγιση ως προς τη λήψη του γεωμετρικού μέσου όρου των προγνωστικών κατανομών που παράγονται από τα δίκτυα εγκατάλειψης. Η διαδικασία της εγκατάλειψης ενεργοποιείται στα πρώτα δύο πλήρως συνδεδεμένα επίπεδα (FC) της Εικόνας 21. Χωρίς τη διαδικασία της εγκατάλειψης, το δίκτυό θα παρουσίαζε σημαντική υπερφόρτωση (Εικόνα 24).



Εικόνα 24: Λειτουργία ενός κανονικού ΝΔ (Αριστερά). ΝΔ με τη διαδικασία της εγκατάλειψης (Δεξιά)

2.4 You Only Look Once (YOLO): Ενιαία Ανίχνευση αντικειμένων σε Πραγματικό Χρόνο

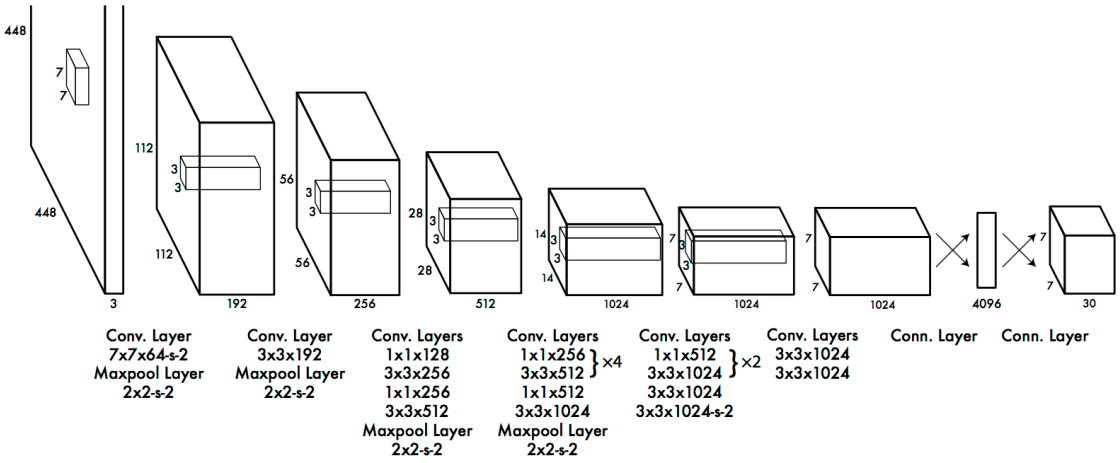
Τα υπάρχοντα συστήματα ανίχνευσης χρησιμοποιούν ταξινομητές για να πραγματοποιήσουν μια ανίχνευση. Για την ανίχνευση ενός αντικειμένου, τα συστήματα αυτά χρησιμοποιούν έναν ταξινομητή, το αξιολογούν μέσα από διαφορετικά επίπεδα, και αποδίδουν διάφορα βάρη στις εικόνες. Συστήματα όπως τα μοντέλα παραμόρφωσης μερών (*Deformable Parts Models: DPM*) χρησιμοποιούν μια προσέγγιση συρόμενων παραθύρων όπου ο ταξινομητής εκτελείται σε ομοιόμορφα τοποθετημένες θέσεις στο σύνολο της εικόνας (P. F. Felzenszwalb, 2010).

Πιο πρόσφατες προσεγγίσεις, όπως η R-CNN μέθοδος δημιουργούν πρώτα δυνητικά πλαίσια οριοθέτησης σε μια εικόνα και στη συνέχεια εκτελείται ένας αλγόριθμος ταξινόμησης σε αυτά τα προτεινόμενα πλαίσια. Μετά τη ταξινόμηση, επανεξετάζονται τα δεδομένα για τη βελτίωση των πλαισίων οριοθέτησης, για την εξάλειψη διπλών ανιχνεύσεων και για την αναδιαμόρφωση των πλαισίων που βασίζονται σε άλλα αντικείμενα (R. Girshick, 2014). Αυτοί οι σύνθετοι αγωγοί είναι αργοί και δύσκολο να βελτιστοποιηθούν επειδή πρέπει κάθε μεμονωμένο στοιχείο να εκπαιδεύεται χωριστά.

Το YOLO είναι ένα ενιαίο συνελικτικό δίκτυο το οποίο ταυτόχρονα προβλέπει πολλαπλά πλαίσια οριοθέτησης (*bounding boxes*) καθώς επίσης και τις πιθανότητες τάξεως γι’ αυτά τα πλαίσια. Το YOLO εκπαιδεύεται με πλήρεις εικόνες και απευθείας βελτιστοποιεί την επίδοση της αναγνώρισης. Αυτό το ενιαίο μοντέλο παρουσιάζει αρκετά πλεονεκτήματα σε σύγκριση με τις παραδοσιακές μεθόδους ανίχνευσης αντικειμένων.

2.4.1 Αρχιτεκτονική του YOLO

Η αρχιτεκτονική του νευρωνικού δικτύου του YOLO περιέχει 106 πλήρη συνελικτικά επίπεδα. Έπειτα το YOLO βελτιώθηκε με διαφορετικές εκδόσεις όπως το YOLOv2 ή YOLOv3 προκειμένου να ελαχιστοποιηθούν τα σφάλματα εντοπισμού και να αυξηθεί η mAP (mean Average Precision: Μέση Ακρίβεια). Στην Εικόνα 25 μπορούμε να δούμε αναλυτικά την αρχιτεκτονική του δικτύου.



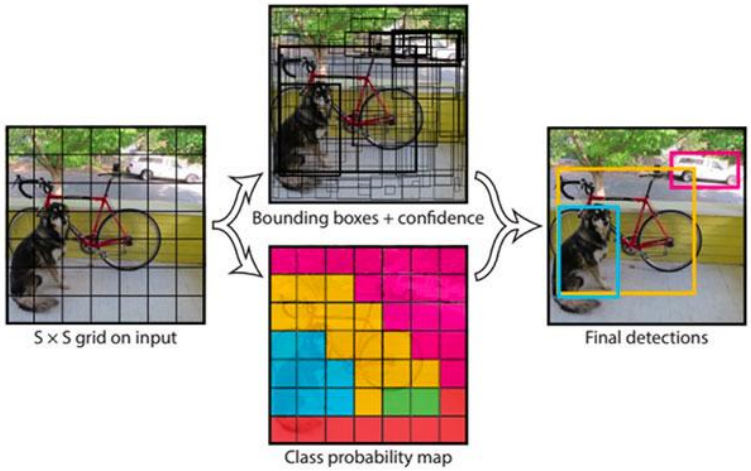
layer	filters	input	output
0 conv	32 3 x 3 / 1	128 x 128 x 3	-> 128 x 128 x 32 0.028 BF
1 conv	64 3 x 3 / 2	128 x 128 x 32	-> 64 x 64 x 64 0.151 BF
2 conv	32 1 x 1 / 1	64 x 64 x 64	-> 64 x 64 x 32 0.017 BF
3 conv	64 3 x 3 / 1	64 x 64 x 32	-> 64 x 64 x 64 0.151 BF
4 Shortcut Layer: 1			
5 conv	128 3 x 3 / 2	64 x 64 x 64	-> 32 x 32 x 128 0.151 BF
6 conv	64 1 x 1 / 1	32 x 32 x 128	-> 32 x 32 x 64 0.017 BF
7 conv	128 3 x 3 / 1	32 x 32 x 64	-> 32 x 32 x 128 0.151 BF
8 Shortcut Layer: 5			
9 conv	64 1 x 1 / 1	32 x 32 x 128	-> 32 x 32 x 64 0.017 BF
10 conv	128 3 x 3 / 1	32 x 32 x 64	-> 32 x 32 x 128 0.151 BF
11 Shortcut Layer: 8			
12 conv	256 3 x 3 / 2	32 x 32 x 128	-> 16 x 16 x 256 0.151 BF
13 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
14 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
15 Shortcut Layer: 12			
16 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
17 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
18 Shortcut Layer: 15			
19 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
20 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
21 Shortcut Layer: 18			
22 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
23 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
24 Shortcut Layer: 21			
25 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
26 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
27 Shortcut Layer: 24			
28 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
29 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
30 Shortcut Layer: 27			
31 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
32 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
33 Shortcut Layer: 30			
34 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
35 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
36 Shortcut Layer: 33			
37 conv	512 3 x 3 / 2	16 x 16 x 256	-> 8 x 8 x 512 0.151 BF
38 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
39 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
40 Shortcut Layer: 37			
41 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
42 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
43 Shortcut Layer: 40			
44 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
45 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
46 Shortcut Layer: 43			
47 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
48 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
49 Shortcut Layer: 46			
50 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
51 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
52 Shortcut Layer: 49			
53 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
54 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
55 Shortcut Layer: 52			

layer	filters	input	output
56 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
57 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
58 Shortcut Layer: 55			
59 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
60 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
61 Shortcut Layer: 58			
62 conv	1024 3 x 3 / 2	8 x 8 x 512	-> 4 x 4 x1024 0.151 BF
63 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
64 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
65 Shortcut Layer: 62			
66 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
67 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
68 Shortcut Layer: 65			
69 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
70 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
71 Shortcut Layer: 68			
72 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
73 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
74 Shortcut Layer: 71			
75 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
76 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
77 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
78 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
79 conv	512 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 512 0.017 BF
80 conv	1024 3 x 3 / 1	4 x 4 x 512	-> 4 x 4 x1024 0.151 BF
81 conv	75 1 x 1 / 1	4 x 4 x1024	-> 4 x 4 x 75 0.002 BF
82 yolo			
83 route	79		
84 conv	256 1 x 1 / 1	4 x 4 x 512	-> 4 x 4 x 256 0.004 BF
85 upsample	2x	4 x 4 x 256	-> 8 x 8 x 256
86 route	85 61		
87 conv	256 1 x 1 / 1	8 x 8 x 768	-> 8 x 8 x 256 0.025 BF
88 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
89 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
90 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
91 conv	256 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 256 0.017 BF
92 conv	512 3 x 3 / 1	8 x 8 x 256	-> 8 x 8 x 512 0.151 BF
93 conv	75 1 x 1 / 1	8 x 8 x 512	-> 8 x 8 x 75 0.005 BF
94 yolo			
95 route	91		
96 conv	128 1 x 1 / 1	8 x 8 x 256	-> 8 x 8 x 128 0.004 BF
97 upsample	2x	8 x 8 x 128	-> 16 x 16 x 128
98 route	97 36		
99 conv	128 1 x 1 / 1	16 x 16 x 384	-> 16 x 16 x 128 0.025 BF
100 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
101 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
102 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
103 conv	128 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 128 0.017 BF
104 conv	256 3 x 3 / 1	16 x 16 x 128	-> 16 x 16 x 256 0.151 BF
105 conv	75 1 x 1 / 1	16 x 16 x 256	-> 16 x 16 x 75 0.010 BF
106 yolo			

Εικόνα 25: Η Αρχιτεκτονική του YOLO

Το δίκτυο χρησιμοποιεί χαρακτηριστικά από ολόκληρη την εικόνα για να προβλέψει κάθε πλαίσιο οριοθέτησης. Επίσης προβλέπει όλα τα πλαίσια οριοθέτησης σε όλες τις κλάσεις για μια εικόνα ταυτόχρονα. Ο σχεδιασμός του YOLO επιτρέπει την εκπαίδευση από άκρο σε άκρο και ταχύτητες σε πραγματικό χρόνο διατηρώντας παράλληλα υψηλή ακρίβεια.

Το σύστημα χωρίζει την εικόνα εισόδου σε ένα δίκτυο *S × S*. Αν το κέντρο ενός αντικειμένου πέσει σε ένα κελί του πλέγματος, αυτό το κελί είναι υπεύθυνο για την ανίχνευση αυτού του αντικειμένου (Εικόνα 26).



Εικόνα 26: Το δίκτυο του YOLO ως μοντέλο οπισθοδρόμησης (S=7).

Κάθε κελί του πλέγματος προβλέπει τα πλαίσια οριοθέτησης *B* και το ποσοστό εμπιστοσύνης (*confidence*) γι' αυτά τα κουτιά. Αυτά τα ποσοστά αξιοπιστίας αντικατοπτρίζουν το πόσο σίγουρο είναι το μοντέλο ότι το πλαίσιο περιέχει ένα αντικείμενο και επίσης πόσο ακριβές θεωρεί το κουτί ότι είναι αυτό που προβλέπει. Τυπικά, η εμπιστοσύνη ορίζεται ως *Pr(Object)**IOU_{pred}^{truth}. Εάν δεν έχει εντοπιστεί κάποιο αντικείμενο σε κάποιο κελί, το ποσοστό εμπιστοσύνης σε αυτό το κελί πρέπει να είναι μηδέν.

Κάθε πλαίσιο οριοθέτησης αποτελείται από 5 προβλέψεις: *x, y, w, h,* και την εμπιστοσύνη. Οι συντεταγμένες (*x, y*) αντιπροσωπεύουν το κέντρο του κουτιού σχετικά με τα όρια του κελιού του δικτύου. Το πλάτος και το ύψος προβλέπονται σε σχέση με ολόκληρη την εικόνα. Τέλος, η πρόβλεψη της εμπιστοσύνης αντιπροσωπεύει το IOU μεταξύ του πλαισίου οριοθέτησης και οποιουδήποτε κιβωτίου του δικτύου. Κάθε κελί του δικτύου προβλέπει επίσης *C* πιθανότητες κλάσης με,

$$C = Pr(Class_i|Object) * IOU_{pred}^{truth}.$$

Αυτές οι πιθανότητες εξαρτώνται από το κελί δικτύου που περιέχει ένα αντικείμενο. Υπολογίζεται μόνο ένα σύνολο πιθανών κλάσεων ανά κελί δικτύου, ανεξάρτητα από τον αριθμό πλαισίων B.

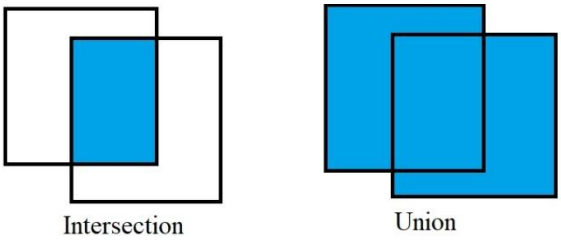
Κατά τη διάρκεια δοκιμής πολλαπλασιάζεται η πιθανότητα τάξης και οι μεμονωμένες προβλέψεις εμπιστοσύνης πλαισίων:

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}},$$

το οποίο μας δίνει ως αποτέλεσμα το διάστημα εμπιστοσύνης της συγκεκριμένης τάξης για κάθε πλαίσιο. Αυτή η τιμή περιλαμβάνει τη πιθανότητα να εμφανίζεται αυτή η τάξη μέσα στο πλαίσιο καθώς και το πόσο καλά το πλαίσιο πρόβλεψης χωράει το αντικείμενο.

2.4.2 Ενοποιημένη Αναγνώριση

Η IOU (Intersection over Union: Τομή επί της Ένωσης) αναπαριστά ένα κλάσμα που παίρνει τιμές [0,1]. Η τομή είναι η περιοχή που επικαλύπτεται μεταξύ του πλαισίου οριοθέτησης που έχει προβλεφτεί και του πλαισίου εμπιστοσύνης και η ένωση είναι η συνολική περιοχή μεταξύ των δύο αυτών όπως απεικονίζεται Στην Εικόνα . Ιδανικά, η IOU πρέπει να είναι κοντά στην τιμή 1, υποδεικνύοντας ότι το προβλεπόμενο πλαίσιο οριοθέτησης τείνει να έχει την μορφή του κελιού δικτύου (Εικόνα 27).



Εικόνα 27:Απεικόνιση της τομής και της ένωσης του πλαισίου οριοθέτησης και του κελιού πλέγματος

Γενικά, το YOLO διαχωρίζει την εικόνα σε πλέγμα *S × S* και για κάθε πλαίσιο του πλέγματος προβλέπει τα εσωτερικά πλαίσια οριοθέτησης *B*, την εμπιστοσύνη γι' αυτά τα πλαίσια, καθώς και τη πιθανότητα τάξης *C*. Αυτές οι προβλέψεις κωδικοποιούνται ως ένας νευρώνας μεγέθους *S × S × (B * 5 + C).* Το YOLO εφαρμόζει την παρακάτω Εξίσωση 11 για να υπολογίσει την απώλεια και τελικά να βελτιστοποιήσει την εμπιστοσύνη:

$$Loss = \lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{A}\mathbb{1}_{ij}^{obj}[(b_{x_i}-b_{\hat{x}_i})^2+(b_{y_i}-b_{\hat{y}_i})^2] \quad +\lambda_{coord}\sum_{i=0}^{s^2}\sum_{j=0}^{A}\mathbb{1}_{ij}^{obj}[(\sqrt{b_{w_i}}-\sqrt{b_{\hat{w}_i}})^2+(\sqrt{b_{h_i}}-\sqrt{b_{\hat{h}_i}})^2]$$
$$+\sum_{i=0}^{s^2}\sum_{j=0}^{A}\mathbb{1}_{ij}^{obj}(C_i-\hat{C}_i)^2 \quad +\lambda_{noobj}\sum_{i=0}^{s^2}\sum_{j=0}^{A}\mathbb{1}_{ij}^{noobj}(C_i-\hat{C}_i)^2+\sum_{i=0}^{s^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}(p_i(c)-\hat{p}_i(c))^2.$$

Εξίσωση 11: Εξίσωση Απώλειας

Πιο αναλυτικά, η λειτουργία απώλειας χρησιμοποιείται για τη διόρθωση του κέντρου και του πλαισίου οριοθέτησης κάθε πρόβλεψης. Κάθε εικόνα χωρίζεται σε ένα S x S πλέγμα, με *A* πλαίσια οριοθέτησης για κάθε πλέγμα. Οι μεταβλητές b_x και b_y αναφέρονται στο κέντρο κάθε πρόβλεψης, ενώ το b_w και b_h αναφέρονται στις διαστάσεις του κιβωτίου οριοθέτησης. Οι μεταβλητές λ_{coord} και λ_{noobj} χρησιμοποιούνται για να δοθεί έμφαση στα πλαίσια που περιέχουν αντικείμενα και να μειωθεί στα πλαίσια χωρίς αντικείμενα. Το *C* αναφέρεται στην εμπιστοσύνη, και το *p(c)* αναφέρεται στη πρόβλεψη ταξινόμησης. Το $\mathbb{1}_{ij}^{obj}$ είναι 1 εάν το j πλαίσιο οριοθέτησης στο i κελί είναι υπεύθυνο για τη πρόβλεψη του αντικειμένου, και 0 σε οποιαδήποτε άλλη περίπτωση. Η απώλεια υποδηλώνει την απόδοση του μοντέλου, όσο μικρότερη είναι η απώλεια τόσο υψηλότερες επιδόσεις έχει.

Όπως αναφέραμε, η απώλεια χρησιμοποιείται για να μετρήσει την απόδοση ενός μοντέλου. Η ακρίβεια των προβλέψεων που πραγματοποιείται από τα μοντέλα κατά την διαδικασία ανίχνευσης αντικειμένων, υπολογίζεται με την εξίσωση μέσης ακρίβειας, όπως παρακάτω:

$$\text{avgPrecision} = \sum_{k=1}^{n} P_{(k)}\Delta r_{(k)}$$

Εξίσωση 12: Εξίσωση Μέσης Ακρίβειας

όπου το *P(k)* αναφέρεται στην ακρίβεια του κατώτερου ορίου *k* ενώ το $\Delta r_{(k)}$ αναφέρεται στην αλλαγή κατά την ανάκληση.

2.4.3 Πρόβλεψη πλαισίου οριοθέτησης αντικειμένου

Το YOLO έχει την ικανότητα να εντοπίζει μια βαθμολογία αντικειμενικότητας για κάθε πλαίσιο οριοθέτησης χρησιμοποιώντας τη λογική παλινδρόμηση. Εάν θεωρήσουμε ότι το πάνω αριστερό κελί έχει μέγεθος c_x, c_y και το πλαίσιο οριοθέτησης έχει ύψος p_hκαι πλάτος p_w , τότε οι τιμές της πρόβλεψης υπολογίζονται σύμφωνα με τα παρακάτω (Εικόνα 28):

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_{w^{e^{t_w}}}$$

$$b_w = p_{w^{e^{t_w}}}$$

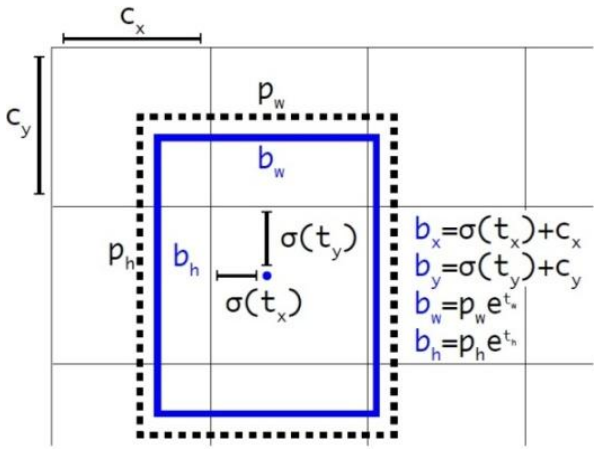
Επομένως, η έξοδος του YOLO για κάθε πλαίσιο είναι της μορφής:

$$y = (c, p_c, b_x, b_y, b_w, b_h,)$$

Όπου η σημασία της κάθε τιμής έχει όπως παρακάτω:

1. Η τιμή c υποδηλώνει το είδος του αντικειμένου που εντοπίστηκε (κατά την εκπαίδευση, το μοντέλο έχει αποδώσει έναν ακέραιο αριθμό για κάθε ξεχωριστή κλάση αντικειμένου π.χ. αυτοκίνητο=1, λεοφωρείο=2,…)
2. Το κέντρο του πλαισίου οριοθέτησης αντικειμένου δίνεται από το σημείο (b_x, b_y)
3. Το πλάτος του πλαισίου είναι b_w
4. Το ύψος του πλαισίου είναι b_h

Επίσης υπολογίζεται και η τιμή p_c η οποία ουσιαστικά είναι η πιθανότητα να υπάρχει ένα αντικείμενο μέσα στο συγκεκριμένο πλαίσιο οριοθέτησης. Στο επόμενο κεφάλαιο θα δούμε τον λόγο που υπολογίζεται αυτή η τιμή.



Εικόνα 28: Πλαίσιο οριοθέτησης με θέση πρόβλεψης. Το πλάτος και το ύψος του πλαισίου υπολογίζονται ως αντισταθμίσεις από τα κεντρομόρια συμπλέγματος. Οι κεντρικές συντεταγμένες του πλαίσιο υπολογίζονται με βάση την σχετική θέση του φίλτρου εφαρμογής χρησιμοποιώντας μία σιγμοειδή συνάρτηση.

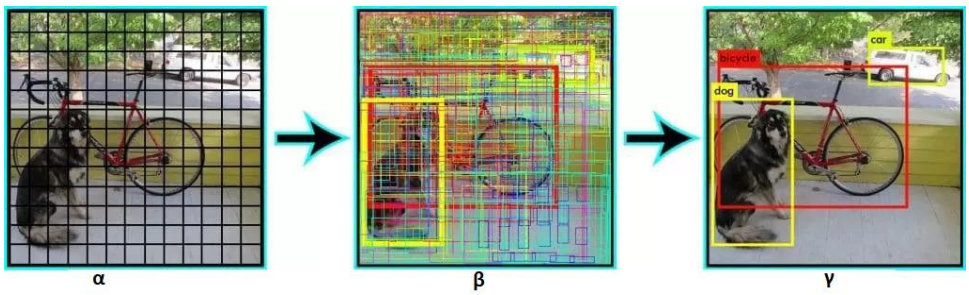
2.4.4 Πως πραγματοποιείται η ανίχνευση

Όπως έχει αναφερθεί, το YOLO χωρίζει την κάθε εικόνα εισόδου σε ένα δίκτυο κελίων S x S (Εικόνα 29α) και κάθε κελί προβλέπει N πλαίσια οριοθέτησης και το διάστημα εμπιστοσύνης. Η εμπιστοσύνη υποδηλώνει την ακρίβεια του πλαισίου οριοθέτησης καθώς και το αν τελικά το πλαίσιο οριοθέτησης περιέχει κάποιο αντικείμενο (ανεξαρτήτως κλάσης). Επίσης υπολογίζει και την βαθμολογία ταξινόμησης για κάθε πλαίσιο και για κάθε κλάση αντικειμένων. Συνδυάζοντας αυτές τις δύο τιμές μπορούμε να υπολογίσουμε την πιθανότητα της κάθε κλάσης να βρίσκεται στο πλαίσιο που προβλέφτηκε.

Τα πλαίσια οριοθέτησης που έχουν υπολογιστεί έχουν λάβει μια τιμή η οποία δεν αναφέρεται στο είδος του αντικειμένου που περιέχεται ή στο διάστημα εμπιστοσύνης αλλά είναι μια τιμή η οποία υποδηλώνει ότι υπάρχει κάτι με μεγάλη σημασία σε αυτό το σημείο. Αν υποθέσουμε ότι θα σχηματίζονταν τα πλαίσια οριοθέτησης με βάση την τιμή αυτή τότε το πάχος του πλαισίου θα ήταν ανάλογο με την τιμή (Εικόνα 29β).

Τέλος, λαμβάνοντας υπόψη την κατανομή των συχνοτήτων καθώς και το ποσοστό εμπιστοσύνης που έχει ορίσει ο χρήστης κατά την εκπαίδευση, εμφανίζονται μόνο τα πλαίσια οριοθέτησης τα οποία έχουν συγκεντρώσει μεγαλύτερο ποσοστό εμπιστοσύνης από αυτό που έχει ορισθεί ως βάση.

Δηλαδή αν το δίκτυό μας χώριζε την εικόνα σε 13 x 13 κελία, υπολόγιζε 5 πλαίσια για κάθε κελί και έχουμε ορίσει ένα ποσοστό εμπιστοσύνης της τάξης του 75%, για να εξάγει το αποτέλεσμα που βλέπουμε στην Εικόνα 29γ το δίκτυο θα είχε πραγματοποιήσει υπολογισμούς σε 13x13x5=845 κελιά και θα μας εμφάνιζε τα πλαίσια οριοθέτησης που θα είχαν ποσοστό εμπιστοσύνης μεγαλύτερο του 75% ($p_c > 0.75$).



Εικόνα 29: Στάδια αναγνώρισης εικόνας μέσω του YOLO

ΚΕΦΑΛΑΙΟ 3

3. Αναγνώριση και Χαρτογράφηση Αντικειμένων με UAV με χρήση CNN

3.1 Εισαγωγή

Η αναγνώριση αντικειμένων αποτελεί πλέον μια συνήθη εργασία της μηχανικής όρασης, και αναφέρεται ουσιαστικά στον προσδιορισμό ή την απουσία συγκεκριμένων χαρακτηριστικών σε δεδομένα εικόνας. Μόλις εντοπιστεί ένα γνωστό χαρακτηριστικό, το αντικείμενο μπορεί να κατηγοριοποιηθεί περεταίρω σε μια από της ορισμένες κλάσεις αντικειμένων. Αυτή η διαδικασία αποτελεί ουσιαστικά την αναγνώριση αντικειμένων. Η ανίχνευση και αναγνώριση αντικειμένων αποτελούν θεμελιώδη δομικά στοιχεία της τεχνητής νοημοσύνης. Μία σημαντική πρόκληση με την ενσωμάτωση της τεχνητής νοημοσύνης και της μηχανής μάθησης σε αυτόνομες λειτουργίες UAV είναι ότι αυτές οι λειτουργείς δεν είναι εκτελέσιμα σε πραγματικό χρόνο ή σε χρόνο κοντά στα πλαίσια του πραγματικού χρόνου, λόγω της πολυπλοκότητας αυτών των εφαρμογών και του μεγάλου κόστους σε υπολογιστική ισχύ. Μία από τις επιλογές που έχουμε είναι η εφαρμογή συστήματος βαθιάς εκμάθησης το οποίο χρησιμοποιεί αλγόριθμο συνελικτικού νευρικού δικτύου για την παρακολούθηση, την ανίχνευση και την ταξινόμηση αντικειμένων από ακατέργαστα δεδομένα του περιβάλλοντος σε πραγματικό χρόνο. Τα τελευταία χρόνια, τα βαθιά συνελικτικά νευρωνικά δίκτυα έχουν δείξει ότι είναι μια αξιόπιστη προσέγγιση για την ανίχνευση αντικειμένων εικόνας και την ταξινόμησή της λόγω της σχετικά υψηλής ακρίβειας και ταχύτητας που διαθέτουν. Επιπλέον, ένας αλγόριθμος CNN επιτρέπει στα UAV που διαθέτουν ένα σύστημα αισθητήρων (στην περίπτωσή μας κάμερα) να μετατρέπουν τις πληροφορίες αντικειμένων από το άμεσο περιβάλλον σε αφηρημένες πληροφορίες που μπορούν να ερμηνευτούν από μηχανές χωρίς ανθρώπινη παρέμβαση. Έτσι για να μπορούμε κατά την διάρκεια μιας πτήσης να πραγματοποιήσουμε αναγνώριση και χαρτογράφηση αντικειμένων σε πραγματικό χρόνο θα πρέπει να μπορούμε να επεξεργαζόμαστε μια ροή δεδομένων (βίντεο) σε πραγματικό χρόνο.

Η διαδικασία της πτήσης μπορεί να χωριστεί σε τρία στάδια. Πρώτον, τα πρωτογενή δεδομένα συλλαμβάνονται με την κάμερα από το UAV κατά τη διάρκεια της πτήσης, σε μορφή βίντεο. Έπειτα πραγματοποιείται η μεταφορά δεδομένων σε μια μονάδα επεξεργασίας στην οποία έχει ενσωματωθεί το CNN. Το τελικό στάδιο αποτελείται από την εξαγωγή του βίντεο με εντοπισμένα και αναγνωρισμένα πλέον τα αντικείμενα καθώς επίσης και ενός αρχείου καταγραφής των θέσεων των αντικειμένων που έχουν αναγνωρισθεί κατά την διάρκεια της πτήσης. Και τα τρία στάδια διεξάγονται σε χιλιοστά του δευτερολέπτου, με αποτέλεσμα να μπορούμε να ισχυριστούμε ότι η αναγνώριση και χαρτογράφηση πραγματοποιείται σχεδόν σε πραγματικό χρόνο. Το κρίσιμο μέρος της διαδικασίας είναι το δεύτερο στάδιο, όπου πραγματοποιείται η μεταφορά των δεδομένων σε εξωτερική μονάδα επεξεργασίας, η ανάλυση και επεξεργασία των δεδομένων εισόδου από την μονάδα επεξεργασίας η οποία ανιχνεύει και ταξινομεί τα περιβάλλοντα αντικείμενα σε πραγματικό χρόνο, και τέλος η επανασύσταση του βίντεο και η μετάδοσή του σε πραγματικό χρόνο. Η όλη διαδικασία μπορεί να λειτουργήσει και με μεμονωμένες φωτογραφίες.

3.2 Δομή του YOLO

Για να μπορέσουμε να αναλύσουμε την δομή του YOLO που χρησιμοποιήθηκε ώστε να πραγματοποιείται η διαδικασία της αναγνώρισης και χαρτογράφησης, θα την χωρίσουμε σε δύο στάδια:

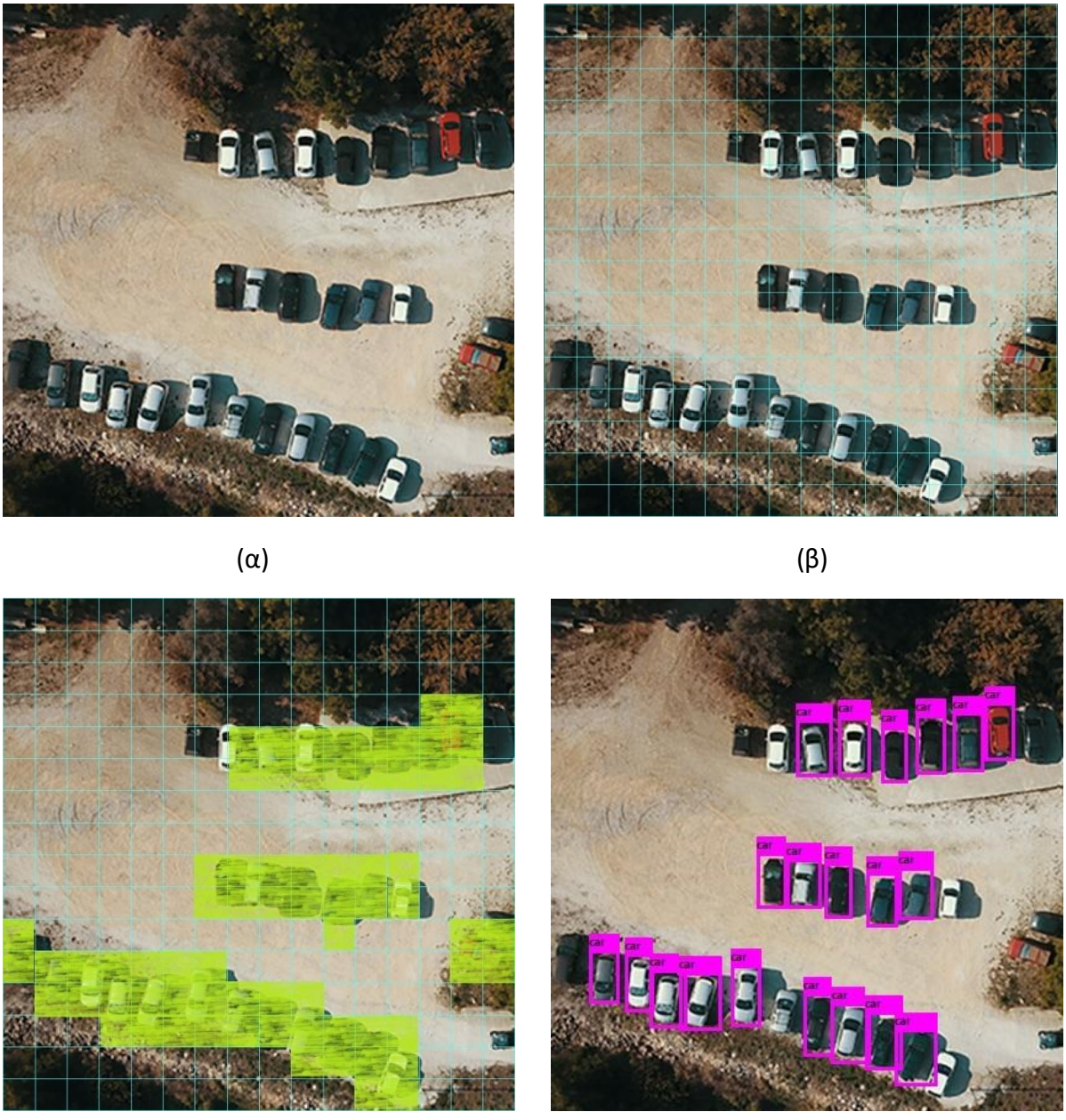
1. Την ανάλυση της αρχιτεκτονικής του δικτύου CNN που εκπαιδεύτηκε και πραγματοποιεί την επεξεργασία των δεδομένων.
2. Την διαδικασία εκπαίδευσης καθώς και τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση.
3. Τα μέσα τα οποία χρησιμοποιήθηκαν για να πραγματοποιηθούν οι δοκιμές.

3.2.1 Ανάλυση της αρχιτεκτονικής του δικτύου CNN.

Ο αλγόριθμος CNN που παρουσιάζεται στην παρούσα έρευνα βασίστηκε σε πρόγραμμα ανοιχτού κώδικα για ανίχνευση αντικειμένων με το YOLO, όπως είδαμε και σε προηγούμενο κεφάλαιο. Το YOLO έχει πολλά πλεονεκτήματα σε σχέση με άλλα παραδοσιακά λογισμικά νευρωνικών δικτύων. Για παράδειγμα, πολλά CNNs χρησιμοποιούν μεθόδους περιφερειακών προτάσεων για να προτείνουν μια δυναμική τοποθέτηση πλαισίων οριοθέτησης μέσα στην εικόνα. Έπειτα, ακολουθεί η ταξινόμηση και η βελτίωση των πλαισίων οριοθέτησης και η εξάλειψη των αντιγράφων. Τελικά, όλα τα πλαίσια οριοθέτησης επαναπροσδιορίζονται με βάση άλλα αντικείμενα που εντοπίζονται. Το ζήτημα με αυτές τις μεθόδους είναι ότι εφαρμόζονται σε πολλαπλές τοποθεσίες και κλίμακες. Οι περιοχές με υψηλή βαθμολογία μιας εικόνας θεωρούνται ως ανίχνευση. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να πληρείται ένα συγκεκριμένο όριο ανίχνευσης. Ενώ αυτοί οι αλγόριθμοι είναι ακριβείς στον εντοπισμό και χρησιμοποιούνται επί του παρόντος σε πολλές εφαρμογές, απαιτούν τεράστια υπολογιστική ισχύ και σχεδόν αδύνατο να βελτιστοποιηθούν. Αυτό τα καθιστά ακατάλληλα για αυτόνομες εφαρμογές UAV που να πραγματοποιούν την επεξεργασία των δεδομένων σε πραγματικό χρόνο. Αφ 'ετέρου, το YOLO χρησιμοποιεί ένα ενιαίο νευρωνικό δίκτυο για να διαιρέσει μια εικόνα σε περιοχές, ενώ προβλέπει την οριοθέτηση πλαισίων και τις πιθανότητες για κάθε περιοχή. Αυτά τα πλαίσια οριοθέτησης σταθμίζονται από τις προβλεπόμενες πιθανότητες. Το κύριο πλεονέκτημα αυτής της προσέγγισης είναι ότι ολόκληρη η εικόνα αξιολογείται από το νευρωνικό δίκτυο, και οι προβλέψεις γίνονται με βάση την έννοια της εικόνας, όχι τις προτεινόμενες περιοχές.

Όπως έχουμε αναφέρει, το YOLO προσεγγίζει την ανίχνευση αντικειμένων ως πρόβλημα παλινδρόμησης. Η διαδικασία ξεκινάει εισάγοντας μια εικόνα στο δίκτυο. Το μέγεθος της εικόνας που εισέρχεται στο δίκτυο πρέπει να είναι σταθερής μορφής. Βάση των δοκιμών που πραγματοποιήθηκαν και των μέσων που διαθέτονταν (μονάδα επεξεργασίας και drone) επιλέχθηκαν οι διαστάσεις 1280 x 720 x 3 για την βέλτιστη απόδοση του συστήματος με βάση την ακρίβεια των αποτελεσμάτων και τον χρόνο εκτέλεσης των διεργασιών αλλά και τιε δυνατότητες των μέσων που χρησιμοποιήθηκαν. Έπειτα ο

αλγόριθμος χωρίζει την εικόνα σε πλαίσια 16 x 16 δημιουργώντας 256 περιοχές διαστάσεων 80x45 (Εικόνα 30β). Για κάθε κελί προβλέπετε το ποσοστό εμπιστοσύνης και τα πλαίσια οριοθέτησης (30γ).



(α) (β)

(γ) (δ)

Εικόνα 30: Απλοποιημένο παράδειγμα επεξεργασίας εικόνας από το Drone.

Σε αυτό το σημείο, κάθε πλαίσιο οριοθέτησης περιέχει τις ακόλουθες πληροφορίες (Εικόνα 31):

$$y = (c, p_c, b_x, b_y, b_w, b_h,)$$

Όπου η σημασία της κάθε τιμής έχει όπως παρακάτω:

1. Η τιμή c υποδηλώνει το είδος του αντικειμένου που εντοπίστηκε (car)
2. Η τιμή p_c είναι η πιθανότητα να περιέχεται αντικείμενο μέσα στο πλαίσιο (%)
3. Το κέντρο του πλαισίου οριοθέτησης αντικειμένου b_x, b_y (left_x,top_y)
4. Το πλάτος του πλαισίου είναι b_w (width)
5. Το ύψος του πλαισίου είναι b_h (height)

```
car: 100%       (left_x:  355   top_y:  438   width:   32   height:   67)
car: 100%       (left_x:  304   top_y:  404   width:   32   height:   67)
car: 100%       (left_x:  704   top_y:  632   width:   72   height:   37)
car: 100%       (left_x:  220   top_y:  427   width:   30   height:   65)
car: 99%        (left_x: 1061   top_y:  343   width:   28   height:   62)
car: 99%        (left_x:  307   top_y:  363   width:   29   height:   61)
car: 97%        (left_x:  610   top_y:    0   width:   51   height:   33)
car: 97%        (left_x:  983   top_y:  263   width:   31   height:   73)
car: 97%        (left_x: 1073   top_y:  307   width:   28   height:   60)
car: 96%        (left_x:  977   top_y:  333   width:   31   height:   58)
car: 96%        (left_x:  875   top_y:  285   width:   31   height:   75)
car: 95%        (left_x: 1001   top_y:  302   width:   28   height:   55)
car: 93%        (left_x: 1153   top_y:  316   width:   28   height:   61)
car: 93%        (left_x: 1232   top_y:  323   width:   36   height:   50)
car: 92%        (left_x:  233   top_y:  464   width:   36   height:   65)
car: 100%       (left_x:   49   top_y:  338   width:   49   height:   81)
```

Εικόνα 31: Απόσπασμα από το αρχείο καταγραφείς του αλγορίθμου

3.2.2 Εκπαίδευση και δεδομένα

Παρόλο που αλγόριθμος του YOLO συνοδεύεται από μια πλατφόρμα ανίχνευσης αντικειμένων και ταξινόμησης, το CNN εξακολουθεί να χρειάζεται εκπαίδευση ώστε ο αλγόριθμος να μάθει να αναγνωρίζει αντικείμενα που θέλουμε εμείς. Το μέγεθος των δεδομένων, η μετατόπιση, ο ρυθμός μάθησης, ο αριθμός των επαναλήψεων και τα κατώτατα όρια ανίχνευσης αποτελούν τις παραμέτρους που τροποποιήθηκαν για την συγκεκριμένη εργασία (παράμετροι που ορίζονται από τον χρήστη). Το δίκτυό μας σχεδιάστηκε για να έχει δομή δικτύου 16 × 16 (S = 16), και εκπαιδεύτηκε σε δύο κατηγορίες αντικειμένων αυτοκίνητο (car) και λεωφορείο (bus).

Για την εκπαίδευση του μοντέλου ακολουθήθηκαν τα παρακάτω βήματα τα οποία αποτελούν και ένα διάγραμμα ροής των ενεργειών που πραγματοποιήθηκαν:

1. Αρχικά θα πρέπει να προσδιορισθούν οι στόχοι που θέλουμε να θέσουμε για το μοντέλο που θέλουμε να δημιουργήσουμε. Ανάλογα με το πλήθος και το είδος των αντικειμένων που θέλουμε να αναγνωρίζει το μοντέλο θα πραγματοποιήσουμε και την ανάλογη συλλογή πληροφοριών. Για κάθε αντικείμενο που θέλουμε να ανιχνεύετε - πρέπει να υπάρχει τουλάχιστον 1 παρόμοιο αντικείμενο στο των σύνολο δεδομένων εκπαίδευσης με περίπου το ίδιο: σχήμα, πλευρά αντικειμένου, σχετικό μέγεθος, γωνία περιστροφής, κλίση και φωτισμό. Έτσι είναι επιθυμητό το σύνολο των δεδομένων εκπαίδευσης να περιλαμβάνει εικόνες με αντικείμενα με διαφορετική: κλίμακα, περιστροφή, φωτισμό, από διαφορετική οπτική γωνία, σε διαφορετικό υπόβαθρο. Ένας γενικός κανόνας είναι ότι θα πρέπει να υπάρχουν τουλάχιστον 500 διαφορετικές εικόνες για κάθε κατηγορία αντικειμένου (ή και περισσότερες αν το αντικείμενο παρουσιάζει ιδιαίτερη μορφή) και θα πρέπει να εκπαιδευτούν τουλάχιστον 2000 επαναλήψεις (iterations) για κάθε κλάση αντικειμένου (4000 στο σύνολο κατ ελάχιστο).
2. Για να είναι δυνατό να μπορέσουμε να μεταφράσουμε τις συντεταγμένες του αρχείου καταγραφής του YOLO σε πραγματικές συντεταγμένες θα πρέπει να πραγματοποιηθεί μια συγκεκριμένη συλλογή δεδομένων ούτως ώστε να υπολογισθεί η αντιστοιχία του μεγέθους 1 εικονοστοιχείου με την πραγματικότητα, δηλαδή πόσα εκατοστά αντιστοιχούν σε ένα pixel.
3. Το επόμενο βήμα είναι η συλλογή των πρωτογενών δεδομένων για τις κλάσης των αντικειμένων που θέλουμε να δημιουργήσουμε. Όπως αναφέρθηκε προηγουμένως, για κάθε αντικείμενο που θέλουμε να ανιχνεύετε πρέπει να υπάρχει τουλάχιστον 1 παρόμοιο αντικείμενο στο των σύνολο δεδομένων εκπαίδευσης με περίπου το ίδιο: σχήμα, πλευρά αντικειμένου, σχετικό μέγεθος, γωνία περιστροφής, κλίση και φωτισμό. Έτσι είναι επιθυμητό το σύνολο των δεδομένων εκπαίδευσης να περιλαμβάνει εικόνες από τα αντικείμενα με διαφορετική: κλίμακα, περιστροφή, φωτισμό, από διαφορετική οπτική γωνία και σε διαφορετικό υπόβαθρο.
4. Για την προετοιμασία του μοντέλου εκπαίδευσης απαιτείται η δημιουργία ενός αρχείου .txt το οποίο θα περιέχει την θέση του κάθε αντικειμένου της φωτογραφίας

με της συντεταγμένες του πλαισίου οριοθέτησης που το περικλείει. Για τον σκοπό αυτό χρησιμοποιήθηκε το εργαλείο YOLO MARK[10] (Εικόνα 32). Με την βοήθεια του εργαλείου αυτού δημιουργείται, για κάθε εικόνα ξεχωριστά, ένα αρχείο .txt με το όνομα της εικόνας, το οποίο για κάθε ένα από τα αντικείμενα της εικόνας δημιουργεί μια νέα γραμμή η οποία έχει τον αριθμό του αντικειμένου και τις συντεταγμένες του, και είναι της μορφής :

<object-class> <x_center> <y_center> <width> <height>

Όπου:

1. <object-class>: Ακέραιος αριθμός των τάξεων που θα χρησιμοποιήσουμε
2. <x_center> <y_center>: το κέντρο του πλαισίου οριοθέτησης που σχηματίζεται
3. <width> <height>: τιμές σχετικές με το μήκος και το ύψος της εικόνας, με τιμή [0.0,1.0]. Υπολογίζονται με τους τύπους: <height> = <absolute_height> / <image_height>, < width > = <absolute_ width > / <image_ width>.

(α) (β)

Εικόνα 32: Το περιβάλλον της εφαρμογής Yolo Mark (α) και το αρχείο .txt που δημιουργεί για κάθε εικόνα (β)

5. Πριν ξεκινήσουμε την εκπαίδευση πραγματοποιήθηκαν οι παρακάτω ρυθμίσεις του αρχείου .cfg του YOLO και προτείνεται να εφαρμοσθούν για την καλύτερη απόδοση του μοντέλου:
    - Ορίζουμε το random=1 (Line:788). Με αυτόν τον τρόπο αυξάνεται η ακρίβεια εκπαιδεύοντας το μοντέλο με διαφορετικές αναλύσεις.
    - Αυξάνουμε την ανάλυση του δικτύου (height=608, width=608 ή οποιασδήποτε τιμής πολλαπλάσιας του 32). Με αυτόν τον τρόπο αυξάνεται η ακρίβεια.

[10] Ιστοσελίδα εφαρμογής: https://github.com/AlexeyAB/Yolo_mark

- Ελέγχουμε το σύνολο δεδομένων αν κάθε αντικείμενο έχει επισημανθεί σωστά - κανένα αντικείμενο στο σύνολο δεδομένων σας δεν πρέπει να είναι χωρίς ετικέτα. Στα περισσότερα προβλήματα εκπαίδευσης - υπάρχουν λάθος ετικέτες στο σύνολο δεδομένων. Γι αυτό τον σκοπό χρησιμοποιούμε και πάλι το YOLO-MARK.
- Είναι επιθυμητό το σύνολο δεδομένων να περιλαμβάνει εικόνες με μη επισημασμένα αντικείμενα που δεν θέλουμε να ανιχνεύονται δηλαδή αρνητικά δείγματα χωρίς οριοθετημένο πλαίσιο (κενά αρχεία .txt). Χρειάζονται περίπου τα μισά αρνητικά δείγματα σε σχέση με το πλήθος των επιθυμητών αντικειμένων.
- Για να εκπαιδευτεί το μοντέλο με εικόνες που περιέχουν πολλά αντικείμενα η κάθε μια, εισάγουμε την παράμετρο max=200 ή μεγαλύτερη τιμή στο τελευταίο [yolo]-layer ή [region]-layer του αρχείου cfg.
- Για την εκπαίδευση του μοντέλου για την ανίχνευση μικρών αντικειμένων (μικρότερα του μεγέθους 16X16 εφόσον το μέγεθος της εικόνας αλλάξει σε 416X416) θέτουμε layers = -1, 11 (Line:720) και stride=4 (Line:717).
- Τροποποίηση του class=80 με τον δικό μας αριθμό κλάσεων (Line:610,696,783).
- Τροποποίηση του filters=255 με την τιμή filters= (classes + 5)x3 (Line:603,689,776). Δηλαδή για classes=1 έχουμε filters=18, για classes=2 έχουμε filters=21 κτλ.
- Για λόγους διευκόλυνσης, το τροποποιημένο αρχείο cfg μπορεί να κατέβει από αυτόν τον σύνδεσμο[11].

6. Συνεχίζουμε την προετοιμασία δημιουργώντας και τοποθετώντας στην σωστή θέση τα παρακάτω αρχεία:
    - Δημιουργία ενός αρχείου obj.names στο build\darknet\x64\data\ με τα ονόματα των αντικειμένων, με το καθένα σε διαφορετική γραμμή.

[11] https://drive.google.com/open?id=17JnhzY2OqkLx9i0jukQyZI0rR8IKoeQn

- Δημιουργία αρχείου obj.data στο build\darknet\x64\data\ με τα παρακάτω στοιχεία (όπου classes= αριθμών των δικών μας αντικειμένων):

  classes= 2

  train = data/train.txt

  valid = data/test.txt

  names = data/obj.names

  backup = backup/

- Τοποθετούμε τα αρχεία εικόνων (.jpg) των αντικειμένων στο build\darknet\x64\data\obj\
- Κατεβάζουμε τα προ-εκπαιδευμένα βάρη για τα συνελικτικά επίπεδα[12] και τα τοποθετούμε build\darknet\x64.

7. Η εκπαίδευση του μοντέλου ξεκινά δίνοντας την παρακάτω εντολή στην γραμμή εντολών:

   darknet.exe detector train data/obj.data yolo-obj.cfg darknet53.conv.74 -map

8. Συνήθως οι 2000 επαναλήψεις (iterations) για κάθε κλάση αντικειμένου είναι αρκετές για την εκπαίδευση του μοντέλου μας. Αλλά για να είμαστε περισσότερο ακριβείς για το πότε πρέπει να σταματήσουμε την εκπαίδευση του μοντέλου μας χρησιμοποιούμε την παρακάτω διαδικασία:

   - Κατά την εκπαίδευση εμφανίζονται διάφορες τιμές στην κονσόλα ελέγχου, και η διαδικασία εκπαίδευσης θα πρέπει να σταματήσει όταν δεν θα μειώνεται πλέον η τιμή **0.XXXXXXX avg**:

     Region Avg IOU: 0.798363, Class: 0.893232, Obj: 0.700808, No Obj: 0.004567, Avg Recall: 1.000000, count: 8 Region Avg IOU: 0.800677, Class: 0.892181, Obj: 0.701590, No Obj: 0.004574, Avg Recall: 1.000000, count: 8

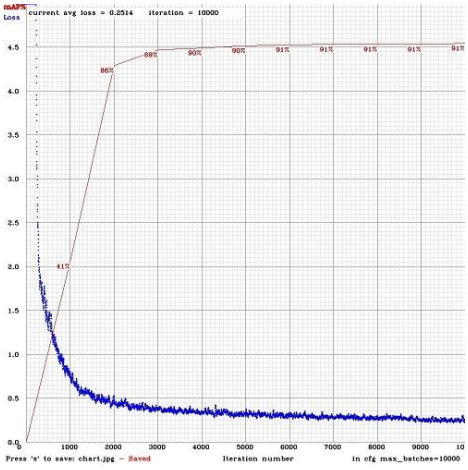
     9002: 0.211667, **0.060730 avg**, 0.001000 rate, 3.868000 seconds, 576128 images Loaded: 0.000000 seconds

     9002: αριθμός της επανάληψης

     0.060730 avg: average loss (σφάλμα). Όσο μικρότερη τιμή τόσο το καλύτερο.

[12] Τα αρχεία βρίσκονται στην ιστοσελίδα: https://pjreddie.com/media/files/darknet53.conv.74

9. Όταν ολοκληρωθεί η εκπαίδευση μπορούμε να βρούμε τα βάρη που δημιουργήθηκαν στον φάκελο darknet\build\darknet\x64\backup.
10. Μέσα στον φάκελο backup εμπεριέχονται αρκετά αρχεία .weights. Το επόμενο βήμα είναι να αξιολογήσουμε τα βάρη με πραγματικά δεδομένα και να κρατήσουμε τα πιο αποτελεσματικά. Για παράδειγμα μπορεί να σταματήσαμε την εκπαίδευση του μοντέλου στις 9000 επαναλήψεις ωστόσο τα καλύτερα αποτελέσματα μπορεί να δίνονται από προηγούμενα βάρη (6000,7000,8000). Αυτό μπορεί να συμβεί λόγο του overfiting όπου είναι η κατάσταση στην οποία ενώ το μοντέλο αναγνωρίζει τα αντικείμενα από το πακέτο δεδομένων με το οποίο εκπαιδεύτικε, δεν είναι σε θέση να αναγνωρίσει αντικείμενα από οποιαδήποτε άλλη εικόνα. Για να μπορέσουμε να αξιολογήσουμε και να συγκρίνουμε τα βάρη που δημιουργήθηκαν χρησιμοποιούμε την συγκρίνουμε την μέση ακρίβεια (mean Average Precision: mAP) του κάθε μοντέλου σύμφωνα με το γράφημα που εξάγει η εφαρμογή. Μεγαλύτερη τιμή mAP σημαίνει και μεγαλύτερη ακρίβεια του μοντέλου. Στην περίπτωσή μας έχουμε στις 8000 επαναλήψεις.



Εικόνα 33: Γράφημα mAP

11. Το τελευταίο στάδιο της εκπαίδευσης είναι και η αξιολόγηση των βαρών που δημιουργήθηκαν με πραγματικά δεδομένα τα οποία συλλέγονται από μια διαφορετική περιοχή με διάφορες τροποποιήσεις στις ρυθμίσεις των εικόνων που θα συλλεχθούν με σκοπό την αξιολόγηση του μοντέλου.

3.3 Μέσα που χρησιμοποιήθηκαν

Τα μέσα που διατέθηκαν για την συλλογή των δεδομένων, την επεξεργασία των δεδομένων και τελικά την δοκιμή του δικτύου που δημιουργήθηκε αποτελούνταν από δύο βασικά στοιχεία: έναν υπολογιστή στον οποίο πραγματοποιήθηκε η προετοιμασία των δεδομένων και η εκπαίδευση του μοντέλου καθώς επίσης αποτελεί και την συσκευή όπου τρέχει ο αλγόριθμος YOLO κατά την διάρκεια της πτήσης και όπου βλέπουμε τα αποτελέσματα, και ένα UAV (στην περίπτωσή μας DJI Mavic Pro). Παρακάτω θα αναλύσουμε τα μέσα που χρησιμοποιήθηκαν καθώς και τα δυνατά σημεία αλλά και τις αδυναμίες τους.

Ο υπολογιστής που χρησιμοποιήθηκε είναι ένας HP 250 G6 με CPU: i5-7200U, RAM:8GB, SSD:500GB και OS: Windows 10. Αποτέλεσε μια ικανοποιητική βάση για την ολοκλήρωση του εγχειρήματος με το μόνο μεγάλο μειονέκτημα ότι δεν διέθετε κάρτα γραφικών συμβατή με CUDA ούτως ώστε να γίνει η εκπαίδευση του μοντέλου σε ένα εύλογο χρονικό διάστημα. Η εκπαίδευση του μοντέλου πραγματοποιήθηκε με βάση την CPU και διήρκησε πάνω από 2 μήνες. Αξίζει να σημειωθεί ότι το χρονικό διάσημα αυτό περιλαμβάνει δοκιμές για να προσδιοριστούν οι βέλτιστες ρυθμίσεις για τον σκοπό μας καθώς επίσης και αποτυχημένες προσπάθειες λόγο διαφόρων εξωγενών παραγόντων (πτώση ρεύματος, σφάλματα δίσκου, σφάλμα λογισμικού κλπ). Ο ρυθμός μάθησης του μοντέλου ήταν της τάξης των 1 ερέθισμα/ 4,5 λεπτά, με αποτέλεσμα να απαιτηθούν περισσότερες από 750 ώρες για να έχουμε τα επιθυμητά αποτελέσματα.

Το UAV που επιλέχθηκε είναι ένα Mavic Pro της DJI. Ο λόγος που επιλέχθηκε το συγκεκριμένο Drone για την εργασία είναι ότι διαθέτη κάμερα με φακό FOV 78.8° 26 mm, f/2.2 και παραμόρφωση < 1,5% καθώς επίσης και ένα γυροσκόπιο πάνω στο οποίο είναι τοποθετημένη η κάμερα με αποτέλεσμα να εξαλείφονται οι κραδασμοί του Drone κατά την πτήση. Αυτό οδηγεί σε μια σταθερή εικόνα βίντεο πράγμα το οποίο επιδρά θετικά στο τελικό αποτέλεσμα. Το μειονέκτημα του Drone είναι ότι ο κατασκευαστής δεν επιτρέπει την ελεύθερη μετάδοση της εικόνας από το Drone προς μια άλλη συσκευή οπότε

απαιτείται μια ιδιαίτερη διαδικασία για να μπορέσουμε να πάρουμε το βίντεο της πτήσης και να το περάσουμε στον αλγόριθμο.

3.4 Αποτελέσματα

Μετά την εκπαίδευση του δικτύου, η ακρίβειά του δοκιμάστηκε σε εικόνες και ροή βίντεο σε πραγματικό χρόνο από πτήση με UAV. Επιπλέον, αξιολογήθηκε η ανίχνευση και αναγνώριση σεναρίων πολλαπλών αντικειμένων. Τα σενάρια πολλαπλών αντικειμένων αναφέρονται στην αναγνώριση περισσότερων από μία κατηγορίας αντικειμένων σε μια δεδομένη εικόνα ή ροή βίντεο. Οι δοκιμές σε πραγματικό χρόνο/πεδίο και τα αποτελέσματα μπορούν να προβληθούν χρησιμοποιώντας αυτόν τον σύνδεσμο (θα ανεβάσω τα βίντεο στο YouTube και θα συμπεριλάβω την σελίδα).

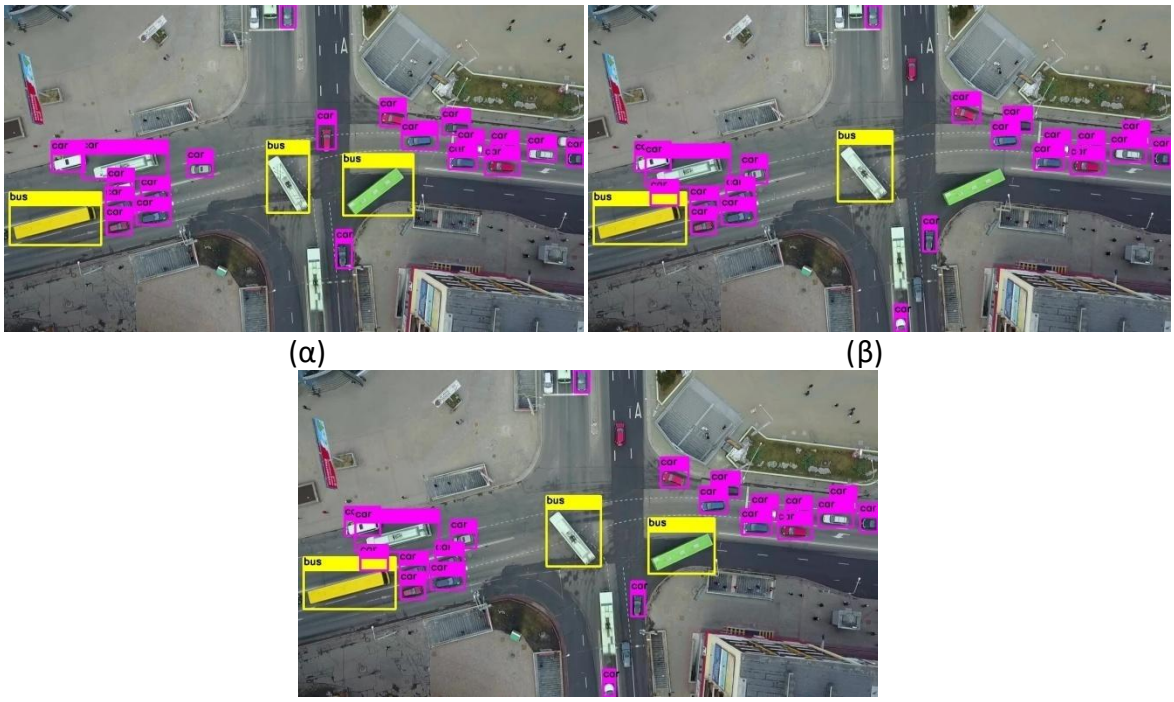
Προκαταρκτικές δοκιμές έδειξαν ότι το CNN ήταν σε θέση να ανιχνεύσει και να εντοπίσει την συγκεκριμένη κλάση σε πραγματικό χρόνο από τη ροή βίντεο που παρέχεται από UAV με ακρίβεια κατά μέσο όρο 75%. Στην Εικόνα 34 εμφανίζεται μια ακολουθία εικόνων από τη ροή βίντεο του UAV στην οποία το CNN μπορεί να εντοπίσει την κλάση αντικειμένων «car». Εδώ μπορούμε να παρατηρήσουμε ότι υπάρχει μια αδυναμία εντοπισμού των αυτοκινήτων με λευκό χρώμα που πιθανών να οφείλετε στην αντίθεση τις εικόνας (εντοπίστηκαν μόνο 3 από τα 6). Με επιπλέον δοκιμές προέκυψε ότι εντοπίζει περίπου το 20-30% των αυτοκινήτων με λευκό χρώμα καταλήγοντας στο γεγονός ότι ευθύνεται η ποιότητα του βίντεο καθώς και η υψηλή αντίθεση. Επίσης, η αναγνώριση των αυτοκινήτων που βρίσκονται στις άκρες της εικόνας δεν είναι ικανοποιητική καθώς υπάρχουν 5 αυτοκίνητα στην Εικόνα 34α και 4 αυτοκίνητα στην 34β που δεν αναγνωρίσθηκαν.

Τέλος στις παρακάτω εικόνες μπορούμε να δούμε στην κάτω σειρά αυτοκινήτων, στην μέση, στο ένα καρέ να μην αναγνωρίζεται ένα ασημένιο αυτοκίνητο αλλά στο επόμενο καρέ να αναγνωρίζεται. Αυτό το γεγονός οφείλεται στην σκιά που δημιουργούν τα αυτοκίνητα αλλά και στο ότι η ποιότητα του βίντεο δεν είναι υψηλή.

(α) (β)

Εικόνα 34: Αναγνώριση με μία κλάση αντικειμένων

Το CNN έχει την ικανότητα να αναγνωρίζει περισσότερες από μία κλάσης. Στην Εικόνα 35 βλέπουμε μια ακολουθία εικόνων από τη ροή βίντεο στην οποία το CNN μπορεί να εντοπίσει και να αναγνωρίζουν δύο είδη αντικειμένων: «car» και «bus». Το CNN ήταν σε θέση να ταξινομήσει και ανιχνεύσει αντικείμενα ακόμη και αν δεν έχουν εμφανιστεί πλήρως στην εικόνα. Για παράδειγμα, στο κάτω, το δεξιό και πάνω τμήμα της Εικόνας 35β, εμφανίζεται μόνο ένα τμήμα από τα αυτοκίνητα που έχουν εντοπισθεί. Ωστόσο αξίζει να σταθούμε στο γεγονός ότι καθώς στο πρώτο καρέ αναγνωρίζονται 3 λεωφορεία (πλαίσιο με κίτρινο χρώμα), στο επόμενο καρέ, επειδή το λεωφορείο άλλαξε κατεύθυνση δεν αναγνωρίσθηκε. Αυτό οφείλεται στο ότι τα δεδομένα εκπαίδευσης δεν περιείχαν κάποιο λεωφορείο το οποίο να εμφάνιζε την ίδια ή έστω παρόμοια κατεύθυνση ώστε να εκπαιδευτεί σωστά το δίκτυο. Το συγκεκριμένο πρόβλημα αποκαταστάθηκε προσθέτοντας επιπλέον εικόνες στο πακέτο δεδομένων και εκπαιδεύοντας το μοντέλο εκ νέου (Εικόνα 35γ). Ένα τελευταίο σημείο το οποίο αξίζει να σταθούμε είναι ότι παρόλο που το μοντέλο εκπαιδεύτηκε εκ νέου και μπόρεσε να αναγνωρίσει το λεωφορείο με την μορφή που αγνοούσε, συνεχίζει να αγνοεί την ύπαρξη του λεωφορείου στο μέσο της εικόνας, καθώς επίσης έκανε και μια λάθος επαναλαμβανόμενη εκτίμηση ενός λεωφορείου ως αυτοκίνητο (Εικόνα 35α,β,γ αριστερό τμήμα).



Εικόνα 35: Αναγνώριση με δύο κλάσης αντικειμένων

3.5 Συντεταγμένες

Σε προηγούμενα κεφάλαια αναλύθηκε το πώς λειτουργεί το CNN σε συνεργασία με το υλικό που επιλέχτηκε να χρησιμοποιηθεί. Όπως είδαμε και στα παραδείγματα όντως, το CNN έχει την ικανότητα να αναγνωρίζει αντικείμενα και να διατηρεί ένα αρχείο καταγραφής όπως αυτό περιγράφεται σε προηγούμενο κεφάλαιο. Ας αναλύσουμε την σημασία των δεδομένων του αρχείου καταγραφής από μια άλλη οπτική.

Έστω ότι κατά την πτήση του UAV η κάμερα είναι στραμμένη κατακόρυφα προς τα κάτω και πραγματοποιεί λήψη μιας φωτογραφίας. Την φωτογραφία αυτή την δίνουμε ως είσοδο στο CNN που εκπαιδεύσαμε. Το αποτέλεσμα του αλγορίθμου φαίνεται στην Εικόνα 35α και το αρχείο καταγραφείς φαίνεται στην Εικόνα 35β. Από τις πληροφορίες της εικόνας του UAV παίρνουμε τις συντεταγμένες του σημείου όπου βρισκόταν την στιγμή που πραγματοποιήθηκε η λήψη της φωτογραφίας. Θεωρητικά, το κέντρο της φωτογραφίας έχει τις ίδιες συντεταγμένες με αυτές της κάμερας. Οι συντεταγμένες της φωτογραφίας είναι: Longitude: 23.26052077, Latitude: 40.76187630 και Altitude: 30.000 (30μ). Οι διαστάσεις της εικόνας είναι γνωστές: 1280 X 720.

Θεωρώντας την φωτογραφία ως ένα επίπεδο κάναβο διαστάσεων 1280 X 720 και ως σημείο (0,0) το πάνω αριστερό σημείο της εικόνας μπορούμε να πούμε ότι το σημείο (640,360) αποτελεί το κέντρο του κανάβου και έχει τις ίδιες συντεταγμένες με την εικόνα με την μόνη διαφορά του υψομέτρου[13].

(α)

```
Total BFLOPS 6.194

 seen 64
 Used AVX
DJI_0085.jpg: Predicted in 0.691964 seconds.
car: 93%        (left_x:  690   top_y:  349   width:  185   height:   96)
car: 87%        (left_x: 1077   top_y:  472   width:  218   height:  122)
```

(β)

Εικόνα 36: Έξοδος του CNN (α) και το αντίστοιχο αρχείο καταγραφής (β)

Το πρώτο αντικείμενο που εντοπίστηκε, σύμφωνα με το αρχείο καταγραφής, βρίσκετε στο σημείο (690,349) ενώ το δεύτερο (1077,472). Οπότε μπορούμε να πούμε με σιγουριά ότι το car1 είναι το κόκκινο αυτοκίνητο και το car2 είναι το μαύρο.

Θεωρώντας ότι πλέον βρισκόμαστε σε ένα καρτεσιανό επίπεδο, μπορούμε να υπολογίσουμε την απόσταση μεταξύ των κέντρων των πλαισίων των αυτοκινήτων με τον παρακάτω τύπο:

$$d = \sqrt{(x_2 - x_2)^2 + (y_2 - y_1)^2}$$

Εξίσωση 13: Εξίσωση εύρεσης απόστασης δύο σημείων

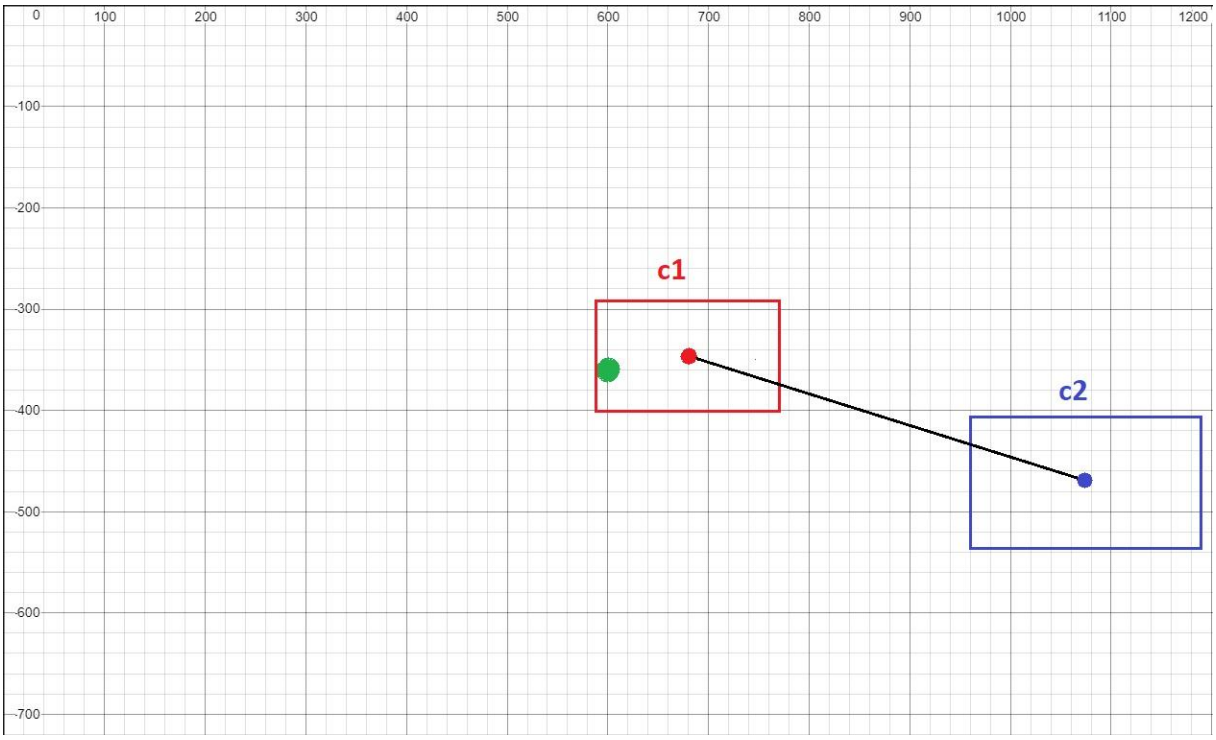
[13] Κατά την πτήση, το Mavic Pro, έχει σχεδιαστεί για να πετάει χρησιμοποιώντας συνδυασμό του βαρόμετρου και του GPS που διαθέτει, ενώ στις ιδιότητες της φωτογραφίας καταγράφει μόνο το υψόμετρο του GPS. Το βαρόμετρο θεωρεί ως 0 το σημείο της απογείωσης ενώ το GPS αντιλαμβάνεται το πραγματικό υψόμετρο. Επομένως το πραγματικό υψόμετρο που βρίσκεται ο δρόμος της φωτογραφίας είναι το υψόμετρο του GPS- την τιμή του βαρόμετρου.

Επομένως έχουμε:

$$d = \sqrt{(1077 - 690)^2 + (472 - 349)^2} = 406$$

Γνωρίζοντας ότι το κόκκινο αυτοκίνητο έχει πραγματικό μήκος 4,320μ και καταλαμβάνει 185 μονάδες (width) συμπεραίνουμε ότι: 1 μονάδα = 23.35 εκ.

Επομένως η πραγματική απόσταση των αυτοκινήτων είναι 406 X 23.35 = 9.48μ (Εικόνα 37).



Εικόνα 37: Αναπαράσταση Εικόνας 35 ως καρτεσιανό επίπεδο

Μια ανάλογη διαδικασία μπορούμε να ακολουθήσουμε για να υπολογίσουμε την απόσταση των κέντρων των αντικειμένων από το κέντρο της εικόνας όπου και έχουμε γνωστές συντεταγμένες. Για να απλοποιήσουμε λίγο την διαδικασία αυτή μπορούμε να δημιουργήσουμε μια εφαρμογή για την αυτοματοποίηση του υπολογισμού. Ένα παράδειγμα της εν λόγο εφαρμογής σε γλώσσα R μπορείτε να το κατεβάσετε από εδώ[14], και είναι:

Library (readtext)

[14] https://drive.google.com/open?id=1vu4WdQpA6ew_w8Cq44484vgfNjyqXnoi

```
df <- read.table("logfile.txt", header = FALSE)
df$apostash_apo_kentro = sqrt((640 - df$V4)^2 + (360 - df$V6)^2)
write.table(df, "dist.txt", sep="\t")
```

Η διαδικασία που ακολουθεί ο αλγόριθμος είναι η εξής:

1. Διαβάζει τα δεδομένα του logfile.txt και δημιουργεί έναν πίνακα τιμών με τα δεδομένα.
2. Για κάθε γραμμή εφαρμόζει την εξίσωση 13 και υπολογίζει την απόσταση του κέντρου του πλαισίου οριοθέτησης του αντικειμένου από το κέντρο της εικόνας σε εκατοστά.
3. Δημιουργεί ένα νέο αρχείο dist.txt το οποίο συμπεριλαμβάνει και την νέα στήλη των δεδομένων με την απόσταση από το κέντρο (Εικόνα 38).

"V2"	"V3"	"V4"	"V5"	"V6"	"V7"	"V8"	"V9"	"V10"	"apostash_apo_kentro"
"car:"	"93%"	"(left_x:"	690	"top_y:"	349	"width:"	185	"height:"	"96)" 51.1957029446808
"car:"	"87%"	"(left_x:"	1077	"top_y:"	472	"width:"	218	"height:"	"122)" 451.12415142619

Εικόνα 38: Αποτελέσματα για τον υπολογισμό της απόστασης

ΚΕΦΑΛΑΙΟ 4

4. Συμπεράσματα

Στην παρούσα έρευνα πραγματοποιήθηκε μια παρουσίαση αλγορίθμων μηχανικής όρασης καθώς και μια μεθοδολογία CNN η οποία αποδεικνύει ότι ακόμα και με χρήση λογισμικού ανοιχτού κώδικα και χωρίς ιδιαίτερα εξεζητημένο εξοπλισμό μπορούμε να επιτύχουμε αναγνώριση και χαρτογράφηση αντικειμένων με UAV. Το YOLO, το οποίο αποτελεί έναν αρκετά γρήγορο, ακριβές και εύκολο στην μετατροπή και εκπαίδευση αλγόριθμο αναγνώρισης αντικειμένων, χρησιμοποιήθηκε ώστε να πραγματοποιηθεί μια πρακτική εφαρμογή αναγνώρισης αντικειμένων με UAV. Το υλικολογισμικό το οποίο χρησιμοποιήθηκε μας επέτρεψε να έχουμε μια ταχύτητα ροής βίντεο της τάξης των 15-20 FPS, ταχύτητα η οποία θα ήταν πολλαπλάσια σε περίπτωση που είχε χρησιμοποιηθεί η GPU αντί της CPU του υπολογιστή. Στην περίπτωση των μεμονωμένων εικόνων η ταχύτητα αναγνώρισης ήταν της τάξης των 0,64 sec για κάθε εικόνα, η οποία αποτελεί μια αρκετά ικανοποιητική ταχύτητα.

Εξηγήσαμε τον τρόπο λειτουργίας του YOLO καθώς και τις τροποποιήσεις που πραγματοποιήθηκαν αλλά και για ποιους λόγους επιλέχθηκαν οι συγκεκριμένες παραμετροποιήσεις. Για να μπορέσουμε να περάσουμε την ροή βίντεο από το Mavic Pro ως είσοδο στο YOLO έπρεπε να καταλάβουμε τον τρόπο με τον οποίο επικοινωνεί το Drone με την συσκευή εξόδου καθώς και την κωδικοποίηση και ανάλυση του βίντεο που παρέχει. Για να το επιτύχουμε αυτό χρησιμοποιήθηκε μια συσκευή Access Point και η δυνατότητα custom share του Drone. Έπειτα η μετατροπή του βίντεο γίνεται OnTheFly για να ταιριάζει με τις απαιτήσεις του YOLO. Η επεξεργασία των δεδομένων εισόδου, η αποθήκευση του αρχείου καταγραφής καθώς και η προβολή σε πραγματικό χρόνο της αναγνώρισης των αντικειμένων πραγματοποιείται στο φορητό υπολογιστή.

Τα δύο μεγάλα εμπόδια που συναντήθηκαν ήταν η εκπαίδευση του μοντέλου και έπειτα η επεξεργασία του μεγάλου όγκου δεδομένων καθώς απαιτούνταν μια τεράστια

υπολογιστική ισχύ για να είμαστε σε θέση να τρέξουμε σε πραγματικό χρόνο την αναγνώριση αντικειμένων σε μια ροή βίντεο με μεγαλύτερη ανάλυση η οποία πιθανώς να μας προσέφερε και περισσότερες πληροφορίες.

Ο αλγόριθμος στην μορφή που έχει τώρα, μπορεί να ανιχνεύσει αντικείμενα σε ροή βίντεο σε ποσοστό μεγαλύτερο του 75% και σε εικόνα σε ποσοστό 80-85 %. Η αστοχία πρόγνωσης ή η λάθος πρόγνωση αντικειμένου οφείλεται κυρίως στην ποιότητα των δεδομένων εισόδου αλλά και στην δομή του YOLO καθώς όπως έχουμε αναφέρει πραγματοποιεί την αναγνώριση μέσων προτύπων που έχει δημιουργήσει ο αλγόριθμος πράγμα που επιφυλάσσει και κάποιες αστοχίες κατά την αναγνώριση. Για χάρη της ταχύτητας επεξεργασίας του αλγόριθμου θυσιάστηκε κομμάτι της ακρίβειας.

Αξίζει να σημειωθεί ότι με βάση το αρχείο καταγραφής και συνδυάζοντας το αρχείο πτήσης του Drone και του βίντεο μπορούμε να πούμε ότι είμαστε σε θέση να καταγράψουμε επακριβώς την θέση των αντικειμένων που αναγνωρίζονται αλλά και να πραγματοποιήσουμε σχετικούς υπολογισμούς μεταξύ των αντικειμένων. Αυτό σημαίνει ότι εκπαιδεύοντας το κατάλληλο μοντέλο, με μια απλή πτήση, μπορούμε να χαρτογραφήσουμε συγκεκριμένα χαρακτηριστικά από μια περιοχή σε ελάχιστο χρόνο.

Μελλοντική εργασία θα αποτελέσει η προσπάθεια να περνάει αυτόματα η γεωγραφική θέση των αντικειμένων που αναγνωρίζονται απευθείας στο αρχείο καταγραφής του YOLO καθώς επίσης και να τοποθετηθεί η μονάδα επεξεργασίας απευθείας πάνω στο UAV και από εκεί να παίρνουμε απευθείας τα αποτελέσματα της χαρτογράφησης.

Βιβλιογραφία

A. Berg, J. D.-F. (2010). *Large scale visual recognition challenge 2010.*

A. Carrio, J. P.-L.-L. (2016). *Ubristes: uav-based building rehabilitation with visible and thermal infrared remote sensing.*

Anderson D. and McNeil, G. (1992). *Artificial Neural Network Technologies: Data Analysis for Software.* Rome.

B.C. Russell, A. T. (2008). *Labelme: a database and web-based tool for image annotation.*

Becker, S. (1991). *Unsupervised learning procedures for neural networks. International Journal of Neural System.*

C. Martinez, C. S. (2014). *Towards autonomous detection and tracking of electric towers for aerial power line inspection.*

D. Cire¸san, U. M. (2012). *Multi-column deep neural networks for image classification.*

Elman, J. L. (1990). *Finding structure in time.*

Fauset, L. (1994). *Foundation of Neural Network.* New Jersey.

foundation, N. N.-A. (1999). *Hykin, S.*

Fukushima, K. (1988). *Neocognitron: a hierarchical neural network.*

G. E. Hinton, S. O.-W. (2006). *A fast learning algorithm for deep belief nets.*

G. Griffin, A. H. (2007). *Caltech-256 object category dataset.*

G.E. Hinton, N. S. (2012). *Improving neural networks by preventing co-adaptation of feature detectors.*

H. Lee, R. G. (2009). *Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.*

Hinton, G. E. (2002). *Training products of experts by minimizing contrastive divergence.*

Hinton, V. N. (2010). *Rectified linear units improve restricted boltzmann machines.*

Hykin, S. (1999). *Neural Network-A comprehensive foundation.*

I. Goodfellow, Y. B. (2016). *Deep Learning.* USA.

Ivakhnenko, A. G. (1971). *Polynomial theory of complex systems IEEE Transactions on Systems, Man and Cybernetics, vol. 1, no.*

J. Deng, W. D. (2009). *ImageNet: a largescale hierarchical image database.* USA.

J. Deng, W. D.-F. (2009). *ImageNet: A Large-Scale Hierarchical Image Database.*

J. Shotton, J. W. (2006). *Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation.*

K. Jarrett, K. K. (2009). *What is the best multi-stage architecture for object recognition?*

Koren, R. B. (2007). *Lessons from the netflix prize challenge.*

Krizhevsky, A. (2010). *Convolutional deep belief networks on cifar-10.* Toronto.

Krizhevsky, A. (2009). *Learning multiple layers of features from tiny images.* Toronto.

Krizhevsky, A. S. (2012). *ImageNet classification with deep convolutional neural networks.*

L. Fei-Fei, R. F. (2007). *Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories.*

L. Li, Y. F. (2016). *Real-time uav weed scout for selective weed control by adaptive robust control and machine learning algorithm.*

Le Lu, Y. Z. (2017). *Deep Learning and Convolutional Neural Networks for Medical Image Computing: Precision Medicine, High Performance and Large-Scale Datasets.*

M. A. Olivares-Mendez, C. F. (2015). *Towards an autonomous vision-based unmanned aerial system against wildlife poachers.*

M. Lin, Q. C. (2013). *Network in network.*

Michael J.A. Berry, G. S. (2004). *Data Mining Techniques for Marketing,Sales, and Customer Relationship Management Second Edition.*

Mitchell, T. M. (1997). *Machine Learning, vol. 45.* USA.

Morton, I. A. (1990). *An introduction to neural computing.* New York: Van Nostrand Reinhold Co.

N. Pinto, D. C. (2008). *Why is real-world visual object recognition hard?*

N. Pinto, D. D. (2009). *A high-throughput screening approach to discovering good forms of biologically inspired visual representation.*

P. F. Felzenszwalb, R. B. (2010). *Object detection with discriminatively trained part based models.*

Perlovsky, L. I. (2001). *Neural Network and Intellect: Model Based Concept.* New York.

R. Girshick, J. D. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation. In Computer Vision and Pattern Recognition.*

Ragav Venkatesan, B. L. (2017). *Convolutional Neural Networks in Visual Computing: A Concise Guide.*

Richardson, M. (2009). *Principal Component Analysis.*

S.C. Turaga, J. M. (2010). *Convolutional networks can learn to generate affinity graphs for image segmentation.*

Sarles, W. R. (1997). *From Neural Network-FAQ ,periodic posting to he usenet news group comp.ai.neural-net. ftp://ftp.sas.com/pub/neural/.*

Sherrah, J. (2017). *Fully Convolutional Networks for Dense Semantic Labelling of High-Resolution Aerial Imagery.*

Smolensky, P. (1986). *Information processing in dynamical systems: foundations of harmony theory.*

VOC2012. *Visual Object Classes Challenge 2012 (VOC2012). Available online: http://host.robots.ox.ac.uk/pascal/.*

Wu, J. (2018). *Convolutional neural networks.*

Y. Bengio, A. C. (2013). *Representation learning: a review and new perspectives.*

Y. Bengio, P. L. (2007). *Greedy layer-wise training of deep networks.*

Y. Le Cun, B. B. (1990). *Handwritten digit recognition with a back-propagation network.*

Y. LeCun, F. H. (2004). *Learning methods for generic object recognition with invariance to pose and lighting.*

Y. LeCun, L. B. (1998). *Gradient-based learning applied to document recognition.*