

ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ/ΒΙΝΤΕΟ ΓΙΑ ΑΝΙΧΝΕΥΣΗ ΖΩΩΝ ΣΕ ΦΥΣΙΚΑ ΤΟΠΙΑ

ANIMAL DETECTION IN NATURAL SCENES THROUGH
IMAGE AND VIDEO PROCESSING

Μαυρόπουλος Ευάγγελος

Διπλωματική εργασία



Πανεπιστήμιο Θεσσαλίας
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Η/Υ

Στην Δήμητρα, στην οικογένειά μου και στους φίλους μου

Επεξεργασία εικόνας/βίντεο για ανίχνευση ζώων σε φυσικά τοπία

Μαυρόπουλος Ευάγγελος

Περίληψη

Οι αλγόριθμοι μηχανικής όρασης δημιουργήθηκαν με σκοπό να δώσουν στους υπολογιστές, αλγοριθμικά, την αίσθηση της όρασης. Η μηχανική όραση βρίσκει εφαρμογή σε ένα μεγάλο εύρος πεδίων όπως η ανίχνευση συμβάντων, η αναγνώριση αντικειμένων, η οργάνωση πληροφοριών και ο έλεγχος διαδικασιών. Στην παρούσα εργασία μελετούμε ένα Σύστημα Ανίχνευσης Ζώων το οποίο στοχεύει τόσο στο υψηλό ποσοστό ανίχνευσης όσο και στην υψηλή υπολογιστική αποδοτικότητα.

Πιο συγκεκριμένα, θα εξετάσουμε μια μέθοδο ανίχνευσης δύο σταδίων. Ο λόγος που χρησιμοποιούμε δύο ταξινομητές (classifiers) είναι η μείωση τόσο των ψευδώς θετικών ανιχνεύσεων, όσο και του χρόνου που απαιτείται για την εκτέλεση πολλών ανιχνεύσεων. Στο πρώτο στάδιο, χρησιμοποιούμε τον αλγόριθμο HOG [1], για την περιγραφή της εικόνας εισόδου, καθώς και έναν ταξινομητή SVM [2] ο οποίος παρέχει στο επόμενο στάδιο περιοχές ενδιαφέροντος, μέσα στις οποίες περιέχονται ζώα και ψευδώς θετικές ανιχνεύσεις. Έπειτα, στο δεύτερο στάδιο, απορρίπτουμε τις ψευδώς θετικές ανιχνεύσεις χρησιμοποιώντας έναν ταξινομητή CNN [3]. Για την εκπαίδευση και τον έλεγχο των αποτελεσμάτων των ταξινομητών χρησιμοποιούμε το σύνολο δεδομένων CIFAR-10 [7].

Η μέθοδος αυτή χρησιμοποιήθηκε για την ανίχνευση ζώων από εικόνες και βίντεο τα οποία λήφθηκαν σε φυσικά τοπία. Για τα πειράματά μας επιλέξαμε να ανιχνεύσουμε το ελάφι. Ο λόγος για αυτή την προτίμηση είναι το πλήθος των ατυχημάτων που συμβαίνουν στο συγκεκριμένο ζώο, είτε σε συγκρούσεις με οχήματα, είτε σε γεωργικές εργασίες.

Τα αποτελέσματα που παρουσιάζονται δείχνουν τις δυνατότητες και τις αδυναμίες ενός τέτοιου συστήματος για χρήση κάτω από πραγματικές συνθήκες καθώς και τα σημεία που επιδέχονται βελτίωση.

Animal detection in natural scenes through image and video processing

Mavropoulos Evangelos

Abstract

Computer vision algorithms have been created to provide computers with an algorithmic sense of vision. These algorithms apply on a wide range of fields such as event detection, object recognition, information retrieval and process control. In this work, we propose an Animal Detection System, aiming at both high detection accuracy and computational efficiency.

Specifically, we propose a two-stage detection method. The reason we employ two machine learning algorithms is the reduction of the false-positive rate and the execution time required for the multi-scale detection procedure of an input image. In the first stage, we use the Histogram of Oriented Gradient (*HOG*) [1] descriptor and a Support Vector Machine (*SVM*) [2] classifier that provides the next stage with a set of regions of interest (ROI) containing target animals and other false positive targets. Subsequently, the second stage, rejects the false positive ROIs by using a Convolutional Neural Network (*CNN*) [3] classifier. To train and evaluate the animal detector, we use CIFAR-10 dataset [7].

The purpose of this method was to detect animals from images and videos captured in natural scenes. For our experiments we used deers as the animal of choice. The reason involving this choice is the high percentage of animal-vehicle collisions incidents that deers are involved.

The results of our experiments indicates the strong and weak points of this system when it is used under real-life conditions, and also some key points that can be optimized.

Περιεχόμενα

| | |
|---|----|
| Κεφάλαιο 1 Εισαγωγή..... | 7 |
| 1.1 Υπόβαθρο και Κίνητρα | 7 |
| 1.2 Στόχοι και Συνεισφορά | 7 |
| 1.3 Διάρθρωση της Διπλωματικής | 8 |
| Κεφάλαιο 2 Υπόβαθρο | 9 |
| 2.1 Ψηφιακή Εικόνα | 9 |
| 2.2 Μετατροπή εικόνων σε χαρακτηριστικά | 9 |
| 2.2.1 Ορισμός χαρακτηριστικών | 10 |
| 2.2.2 Περιγραφή χαρακτηριστικών..... | 10 |
| 2.3 Υπολογιστική Όραση | 11 |
| 2.3.1 Μηχανική Μάθηση | 11 |
| 2.3.2 Σύνολα Δεδομένων | 12 |
| 2.4 Σχετικές εργασίες | 12 |
| Κεφάλαιο 3 Σχεδίαση και Υλοποίηση του Συστήματος | 13 |
| 3.1 Ανάλυση της Αρχιτεκτονικής του Συστήματος | 13 |
| 3.1.1 Ανάγκη ύπαρξης Περιγραφέα | 13 |
| 3.1.2 Ανάγκη ύπαρξης δύο σταδίων ταξινόμησης | 13 |
| 3.2 Περιγραφέας HOG | 14 |
| 3.3 Ταξινόμηση με χρήση SVM..... | 18 |
| 3.3.1 Ανάλυση αλγορίθμου SVM | 18 |
| 3.3.2 Υλοποίηση του αλγορίθμου SVM..... | 20 |
| 3.4 Ταξινόμηση με χρήση CNN | 20 |
| 3.4.1 Νευρωνικά Δίκτυα | 20 |
| 3.4.2 Συνελκτικά Νευρωνικά Δίκτυα..... | 23 |
| 3.4.3 Υλοποίηση του ταξινομητή CNN | 24 |
| Κεφάλαιο 4 Αποτελέσματα | 25 |
| 4.1 Εκπαίδευση αλγορίθμων..... | 25 |
| 4.1.1 Εκπαίδευση SVM..... | 25 |
| 4.1.2 Εκπαίδευση CNN | 27 |
| 4.2 Πείραμα με σύνολο δεδομένων ελέγχου CIFAR-10 | 28 |
| 4.2.1 Μελέτη ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM | 29 |

| | |
|---|----|
| 4.2.2 Μελέτη ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή CNN | 30 |
| 4.2.3 Μελέτη ευαισθησίας κατωφλίου για το συνολικό σύστημα με χρήση CNN 2 κλάσεων | 31 |
| 4.2.4 Μελέτη ευαισθησίας κατωφλίου για το συνολικό σύστημα με χρήση CNN 10 κλάσεων | 32 |
| 4.3 Πείραμα με τυχαίες εικόνες | 33 |
| 4.4 Χρόνοι εκτέλεσης | 35 |
| 4.4.1 Ανάλυση χρόνων για κλιμακωτή ανίχνευση | 35 |
| 4.4.2 Ομαδοποίηση θετικών ανιχνεύσεων του 1 ^{ου} σταδίου | 37 |
| Κεφάλαιο 5 Συμπεράσματα και μελλοντικοί στόχοι | 38 |
| Βιβλιογραφία/Αναφορές | 39 |

Κατάλογος εικόνων

| | |
|--|----|
| Εικόνα 2.1 Ψηφιακή εικόνα (Αριστερά) και τα διακριτά ρixel που προκύπτουν από την μεγέθυνση (Δεξιά) | 9 |
| Εικόνα 2.2 Περιγραφέας Histogram of Oriented Gradients (HOG). Αριστερά η αρχική εικόνα και δεξιά τα χαρακτηριστικά της. [28] | 11 |
| Εικόνα 3.1 Παράδειγμα ολισθαίνωντος παραθύρου. [28]..... | 15 |
| Εικόνα 3.2 Ιστόγραμμα κλίσεων [28]..... | 16 |
| Εικόνα 3.3 Αναπαράσταση των μπλοκ και της αλληλοεπικάλυψής τους. [28]..... | 16 |
| Εικόνα 3.4 Παραδείγματα εφαρμογής του περιγραφέα HOG. [28]..... | 17 |
| Εικόνα 3.5 Παράδειγμα διαχωριστικών υπερεπιπέδων στον δισδιάστατο χώρο. [29]..... | 18 |
| Εικόνα 3.6 Βέλτιστο υπερεπίπεδο [29]..... | 19 |
| Εικόνα 3.7 Παράδειγμα MLP 3 επιπέδων, με 3 εισόδους, 2 εξόδους και 1 κρυφό επίπεδο 5 νευρώνων. [31]..... | 21 |
| Εικόνα 3.8 Παράδειγμα άθροισης των εισόδων στον νευρώνα [31]..... | 22 |
| Εικόνα 3.9 Αναπαράσταση του συνελκτικού νευρωνικού δικτύου LeNet. [30] | 23 |
| Εικόνα 3.10 Αναπαράσταση συνελκτικού νευρωνικού δικτύου μαζί με τεχνητό νευρωνικό δίκτυο [30] | 24 |
| Εικόνα 4.1 Μετρικές κατά την εκπαίδευση του γραμμικού SVM | 26 |
| Εικόνα 4.2 Σιγμοϊδής συνάρτηση μεταβολής του ρυθμού μάθησης κατά την εκπαίδευση του CNN..... | 27 |
| Εικόνα 4.3 Μετρικές κατά την εκπαίδευση του CNN CIFAR-10.(1 epoch = 50 iterations)..... | 28 |
| Εικόνα 4.4 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM..... | 29 |
| Εικόνα 4.5 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM..... | 30 |
| Εικόνα 4.6 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για ολόκληρο το σύστημα με χρήση CNN 2 κλάσεων. | 31 |
| Εικόνα 4.7 Πείραμα κατωφλίου για ολόκληρο το σύστημα Πείραμα κατωφλίου για ολόκληρο το σύστημα με χρήση CNN 10 κλάσεων..... | 32 |
| Εικόνα 4.8 Precision-Recall στο πείραμα τυχαίων εικόνων. | 34 |
| Εικόνα 4.9 Τιμές Precision και Recall ως προς την τιμή του κατωφλίου..... | 34 |
| Εικόνα 4.11 Ανιχνεύσεις συστήματος σε νυχτερινή φωτογραφία..... | 35 |
| Εικόνα 4.11 Ανίχνευση συστήματος σε περιοχές με βλάστηση..... | 35 |

Κεφάλαιο 1 Εισαγωγή

1.1 Υπόβαθρο και Κίνητρα

Η πιο γνωστή χρήση Προηγμένων Συστημάτων Υποβοήθησης Οδηγού έως τώρα είναι η αναγνώριση οδικών σημάτων και η ανίχνευση πεζών. Όμως οι συγκρούσεις οχημάτων με ζώα είναι ένα φαινόμενο που αποκτά όλο και μεγαλύτερες διαστάσεις, όχι μόνο εξαιτίας της αύξησης των οχημάτων αλλά και της επέκτασης των οδικών αρτηριών.

Σε χώρες με μεγάλο πληθυσμό αγρίων ζώων και αυτοκινητόδρομους οι οποίοι διέρχονται μέσα από δάση, οι πιθανότητες σύγκρουσης οχημάτων με άγρια ζώα είναι ιδιαίτερα μεγάλες. Αυτό έχει σαν αποτέλεσμα θανάτους, τραυματισμούς και οικονομικές ζημιές λόγω των καταστροφών που προκαλούνται. Σύμφωνα με έρευνα του Καναδικού Υπουργείου Μεταφορών [4], κάθε ώρα συμβαίνουν 4 με 8 συγκρούσεις οχημάτων με άγρια ζώα στον Καναδά. Στις Η.Π.Α παρατηρούνται περισσότερα από δύο εκατομμύρια τέτοια συμβάντα κάθε χρόνο τα οποία έχουν σαν αποτέλεσμα 200 θανάτους ανθρώπων, 29000 τραυματισμούς και καταστροφές περιουσίας συνολικού κόστους μεγαλύτερου των 1.1 δις δολαρίων [5].

Για την αντιμετώπιση αυτού του προβλήματος οι νέες γενιές αυτοκινήτων είναι εξοπλισμένες με Προηγμένα Συστήματα Υποβοήθησης Οδηγού τα οποία είναι ικανά να ανιχνεύουν ζώα που εμφανίζονται μπροστά στα οχήματα και να προειδοποιούν τον οδηγό για τον κίνδυνο σύγκρουσης [26]. Ταυτοχρόνως, αντίστοιχα Συστήματα Ανίχνευσης Ζώων μπορούν να τοποθετούνται σε περάσματα και να ενημερώνουν τους οδηγούς για την ύπαρξη ζώων στο δρόμο. Ειδικότερα, για τα μη επανδρωμένα οχήματα που ελέγχονται από κάποια τεχνητή νοημοσύνη, η ύπαρξη Συστημάτων Ανίχνευσης Ζώων είναι υποχρεωτική.

Οι καταστροφές που προκαλούν τα ζώα στον τομέα της γεωργίας είναι εξίσου σημαντικές. Κατά την περίοδο του θερισμού τα νεογέννητα ελάφια τα οποία κρύβονται μέσα στους αγρούς πέφτουν συχνά θύματα των αλωνιστικών μηχανών. Υπολογίζεται ότι το 25% των νεογέννητων ελαφιών σκοτώνονται από αυτή διαδικασία του θερισμού [6]. Εκτός της ευημερίας των ζώων, υπάρχει ένας ακόμα λόγος για την αποφυγή αυτών των συμβάντων. Σε περίπτωση που το πτώμα του ζώου περάσει απαρατήρητο, η σήψη του δημιουργεί τοξίνες οι οποίες καθιστούν τα σιτηρά ακατάλληλα για κατανάλωση.

Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιούνται μη επανδρωμένα αεροσκάφη, τα οποία εντοπίζουν τα ζώα που βρίσκονται θέση κινδύνου. Με αυτό τον τρόπο βοηθούν στην απομάκρυνση τους και στην αποφυγή ατυχημάτων [6].

1.2 Στόχοι και Συνεισφορά

Στην παρούσα εργασία μελετούμε ένα Σύστημα Ανίχνευσης Ζώων το οποίο δέχεται σαν είσοδο εικόνες, που προέρχονται από συσκευές καταγραφής εικόνας (2D κάμερα) όπως φαίνεται στο Σχήμα 1.1. Στην εικόνα εισόδου γίνεται εξαγωγή χαρακτηριστικών σε διανύσματα μέσω του αλγορίθμου HOG [1] με σκοπό την μείωση του μεγέθους της εισόδου του ταξινομητή και την βελτίωση της ανθεκτικότητας τον

αποτελεσμάτων του σε μικρές αλλαγές της εισόδου. Τα διανύσματα που παράγονται, δίνονται σαν είσοδο στον ταξινομητή SVM [2] ο οποίος σαρώνει την εικόνα με την χρήση ενός παραθύρου, κάνοντας μια γρήγορη αλλά όχι τόσο αποτελεσματική κατηγοριοποίηση, και ταξινομεί τα παράθυρα σαν σημεία ενδιαφέροντος ή όχι. Οι περιοχές της αρχικής εικόνας εισόδου που χαρακτηρίστηκαν ως σημεία ενδιαφέροντος παρέχονται στον ταξινομητή CNN [3], του οποίου σκοπός είναι η απομάκρυνση των λανθανόντων θετικών ανιχνεύσεων. Οι περιοχές οι οποίες θα χαρακτηριστούν και από τον δεύτερο ταξινομητή ως σημεία ενδιαφέροντος δίνονται ως έξοδος από το σύστημα.



Σχήμα 1.1 Τα στάδια του συστήματος

Για την εκπαίδευση των ταξινομητών SVM και CNN χρησιμοποιούμε το σύνολο δεδομένων CIFAR-10 [7] και κάποιες εικόνες οι οποίες περιέχουν φυσικά τοπία από το σύνολο δεδομένων [32]. Οι αλγόριθμοι που χρησιμοποιούμε περιλαμβάνονται στις βιβλιοθήκες OpenCV [8] και Caffe [9] και χρησιμοποιούν την GPU για την εκτέλεση των πιο υπολογιστικά απαιτητικών υπολογισμών τους.

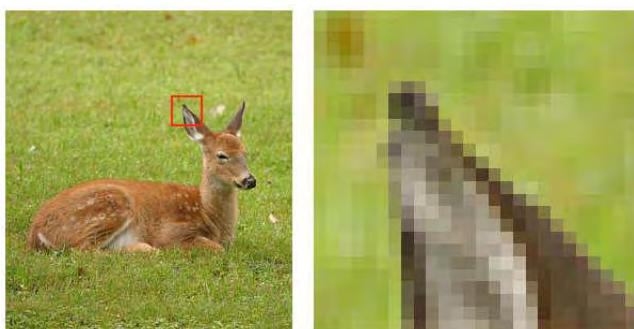
1.3 Διάρθρωση της Διπλωματικής

Η παρούσα διπλωματική οργανώνεται ως εξής. Στο κεφάλαιο 2, γίνεται η εισαγωγή στις βασικές έννοιες που απαιτούνται για την κατανόηση των τεχνικών που αναλύονται στα επόμενα κεφάλαια. Επίσης πραγματοποιείται μια ανασκόπηση σε αντίστοιχες έρευνες που έχουν γίνει πάνω στη ανίχνευση ζώων. Στο κεφάλαιο 3 ακολουθεί η περιγραφή της υλοποίησης του συστήματος που μελετούμε, αναλύοντας τους αλγόριθμους HOG [1], SVM [2] και CNN [3]. Ακολούθως στο κεφάλαιο 4 αναλύουμε την προετοιμασία των πειραμάτων και παραθέτουμε τα αποτελέσματα που πήραμε από αυτά. Κλείνοντας στο κεφάλαιο 5 παραθέτουμε τα συμπεράσματα που αντλήσαμε από τα πειράματα και αναφέρουμε πιθανούς μελλοντικούς στόχους.

Κεφάλαιο 2 Υπόβαθρο

2.1 Ψηφιακή Εικόνα

Η ψηφιακή εικόνα μπορεί να οριστεί ως μια δισδιάστατη συνάρτηση $f(x, y)$, όπου x και y αποτελούν τις χωρικές συντεταγμένες και η τιμή της συνάρτησης f αποτελεί την ένταση ή το επίπεδο γκρίζου στο σημείο αυτό. Κάθε σημείο (x, y) ονομάζεται εικονοστοιχείο. Για τις ασπρόμαυρες εικόνες όσο πιο φωτεινό είναι ένα σημείο (x, y) τόσο μεγαλύτερη θα είναι η τιμή της f , ενώ για μικρές τιμές της f η εικόνα πλησιάζει το μαύρο. Για τις έγχρωμες εικόνες η απόχρωση κάθε πιξελ εξαρτάται από τον συνδυασμό τριών διαφορετικών τιμών, η κάθε μια εκ των οποίων αντιστοιχεί σε τρία διαφορετικά χρωματικά κανάλια (RGB, YCbCr, HSV, CMYK). Για το πρώτο στάδιο του συστήματος χρησιμοποιούμε ασπρόμαυρες εικόνες και για το δεύτερο στάδιο έγχρωμες.



Εικόνα 2.1 Ψηφιακή εικόνα (Αριστερά) και τα διακριτά pixel που προκύπτουν από την μεγέθυνση (Δεξιά)

2.2 Μετατροπή εικόνων σε χαρακτηριστικά

Στους τομείς της υπολογιστικής όρασης και της ψηφιακής επεξεργασίας εικόνας ορίζουμε σαν χαρακτηριστικό (*feature*) ένα κομμάτι πληροφορίας που σχετίζεται με μια συγκεκριμένη εφαρμογή. Τα χαρακτηριστικά μπορεί να αποτελούνται από συγκεκριμένες δομές της εικόνας όπως σημεία, απότομες αλλαγές στην φωτεινότητα ή αντικείμενα.

Τα χαρακτηριστικά αυτά αποτελούν έναν οπτικό περιγραφέα (*visual descriptor*) της εικόνας. Οι οπτικοί περιγραφείς χωρίζονται σε δύο βασικές κατηγορίες:

- Περιγραφείς γενικών πληροφοριών οι οποίοι περιέχουν πληροφορίες χαμηλού επιπέδου όπως χρώμα, υφή, σχήμα και κίνηση.
- Περιγραφείς πληροφοριών συγκεκριμένου τομέα οι οποίοι περιέχουν πληροφορίες υψηλού επιπέδου όπως αντικείμενα και γεγονότα που υπάρχουν ή συμβαίνουν στην

εικόνα. Ένα χαρακτηριστικό παράδειγμα τέτοιου περιγραφέα είναι η αναγνώριση προσώπου.

2.2.1 Ορισμός χαρακτηριστικών

Δεν υπάρχει συγκεκριμένος ορισμός που να περιγράφει την έννοια του χαρακτηριστικού. Η έννοια αυτή ορίζεται και πλάθεται ανάλογα με τις ανάγκες ή τα προβλήματα της κάθε εφαρμογής. Όμως υπάρχουν κάποιοι τύποι χαρακτηριστικών οι οποίοι χρησιμοποιούνται συχνά:

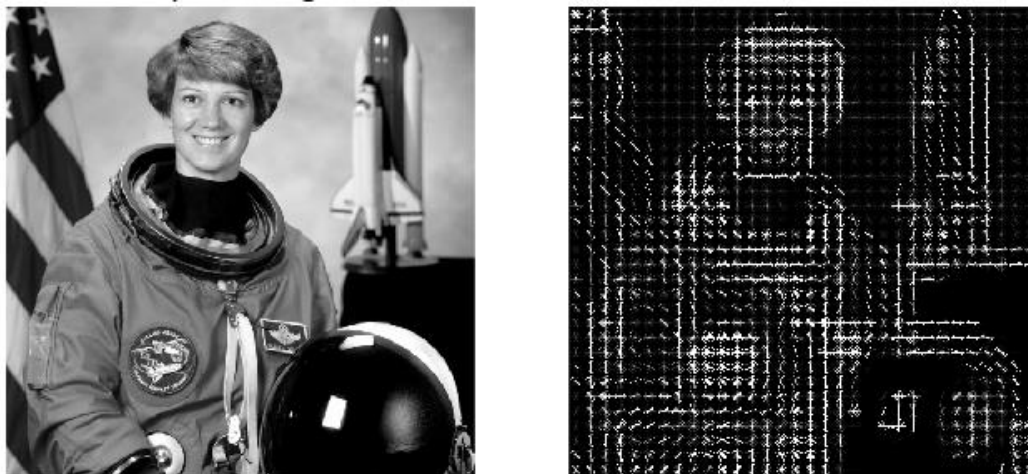
- Το χρώμα, το πιο βασικό χαρακτηριστικό του οπτικού περιεχομένου
- Η υφή των επιφανειών που υπάρχουν σε μια εικόνα
- Το σχήμα των αντικειμένων που περιέχονται σε μια εικόνα
- Η κατεύθυνση της κίνησης των αντικειμένων που εικονίζονται σε μια ακολουθία εικόνων
- Η τοποθεσία αντικειμένων με σημείο αναφοράς τις χωρικές συντεταγμένες της εικόνας. Η τοποθεσία μπορεί να προσδιοριστεί και με σημείο αναφοράς τις χρονικές συντεταγμένες της εικόνας.

2.2.2 Περιγραφή χαρακτηριστικών

Στον τομέα της μηχανικής εκμάθησης (*machine learning*), της αναγνώρισης προτύπων (*pattern recognition*) και της επεξεργασίας εικόνας (*image processing*) η περιγραφή χαρακτηριστικών είναι μια συχνά χρησιμοποιούμενη τεχνική η οποία δίνει λύσεις σε διάφορα προβλήματα.

Ένα από αυτά τα προβλήματα είναι ο μεγάλος όγκος δεδομένων που χρειάζεται να δοθούν ως είσοδος σε έναν αλγόριθμο μηχανικής εκμάθησης για την διαδικασία της εκπαίδευσης. Όταν αυτά τα δεδομένα απαρτίζονται από εικόνες, μια λύση σε αυτό το πρόβλημα είναι η μείωση του μεγέθους της εικόνας κάνοντας χρήση παρεμβολής. Η λύση αυτή όμως μειώνει την πληροφορία που περιέχεται στην εικόνα. Αντίθετα η χρήση τεχνικών περιγραφής χαρακτηριστικών μπορεί προσφέρει μια διαφορετική απεικόνιση της εικόνας, η οποία θα απαιτεί λιγότερα δεδομένα για την περιγραφή της, ελαχιστοποιώντας ταυτόχρονα την απώλεια πληροφορίας. Αυτό επιτυγχάνεται διατηρώντας την πληροφορία που θεωρείται απαραίτητη για τον εκάστοτε σκοπό και απορρίπτοντας την πληροφορία η οποία είτε είναι ευαίσθητη στον θόρυβο(απόχρωση) είτε δεν μπορεί να συνεισφέρει στην επίτευξη του σκοπού μας(υφή).

Ένα άλλο πρόβλημα είναι έλλειψη ανθεκτικότητα των αλγορίθμων μηχανικής εκμάθησης σε μικρές αλλαγές. Εάν τα δεδομένα που δίνονται ως είσοδος σε ένα τέτοιο αλγόριθμο είναι εικόνες, μικρές αλλαγές στον φωτισμό της εικόνας μπορούν να οδηγήσουν σε λανθασμένες κατηγοριοποιήσεις. Απαιτείται μια λύση η οποία να μην επηρεάζεται από τέτοιες μικρές μεταβολές. Τέτοια λύση είναι η αξιοποίηση των χαρακτηριστικών που γνωρίζουμε ότι είναι ανθεκτικές στις συγκεκριμένες αλλοιώσεις.



Εικόνα 2.2 Περιγραφέας Histogram of Oriented Gradients (HOG). Αριστερά η αρχική εικόνα και δεξιά τα χαρακτηριστικά της. [28]

2.3 Υπολογιστική Όραση

Η υπολογιστική όραση (*computer vision*) είναι ένα επιστημονικό πεδίο της τεχνητής νοημοσύνης το οποίο επιχειρεί να αναπαράγει αλγοριθμικά την αίσθηση της όρασης, για ηλεκτρονικούς υπολογιστές και ρομποτικές εφαρμογές. Τα συστήματα μηχανικής όρασης συνδυάζουν ψηφιακή επεξεργασία εικόνων και μηχανική μάθηση με σκοπό την ανάπτυξη της δυνατότητας κατηγοριοποίησης εικόνων.

2.3.1 Μηχανική Μάθηση

Η μηχανική μάθηση (*machine learning*) είναι ένα επιστημονικό πεδίο της τεχνητής νοημοσύνης το οποίο περιλαμβάνει αλγόριθμους και μεθόδους που επιτρέπουν στους υπολογιστές να μαθαίνουν. Τα συστήματα μηχανικής μάθησης έχουν την δυνατότητα να λειτουργούν με βάση την αυτοματοποιημένη ανάλυση συνόλων δεδομένων και όχι την διαίσθηση των μηχανικών που τα προγραμματίσαν. Οι αλγόριθμοι αυτοί κατηγοριοποιούνται σε δύο μεγάλες ομάδες:

- Αλγόριθμοι επιτηρούμενης μάθησης (*supervised learning*) οι οποίοι κατασκευάζουν ένα μοντέλο για κάποιο σύνολο εισόδων, γνωρίζοντας εξαρχής τις ορθές εξόδους. Σκοπός είναι η γενίκευση του μοντέλου αυτού για εισόδους με άγνωστη έξοδο.
- Αλγόριθμοι μη επιτηρούμενης μάθησης (*unsupervised learning*) οι οποίοι κατασκευάζουν ένα μοντέλο για κάποιο σύνολο εισόδων χωρίς να γνωρίζουν τις επιθυμητές εξόδους.

Στην περίπτωση των συστημάτων υπολογιστικής όρασης η είσοδος αποτελείται από εικόνες ή χαρακτηριστικά εικόνων και η έξοδος είναι η κατηγορία που υπάγεται η εικόνα εισόδου.

Οι πιο διαδεδομένοι αλγόριθμοι επιτηρούμενης μάθησης είναι οι:

- Λογιστική παλινδρόμηση (*Logistic Regression*) [10]
- Μηχανές διανυσμάτων υποστήριξης (*Support Vector Machines*) [2]
- Νευρωνικά δίκτυα *Neural Networks* [11]

Οι πιο διαδεδομένοι αλγόριθμοι μη επιτηρούμενης μάθησης είναι οι:

- Κ-κοντινότερων γειτόνων (*k-nearest neighbors*) [12]
- Κ-μέσων (*k-means*) [13]

2.3.2 Σύνολα Δεδομένων

Τα σύνολα δεδομένων είναι συλλογές από δεδομένα, κατηγοριοποιημένα και μη, τα οποία χρησιμοποιούνται από τους αλγόριθμους μηχανικής μάθησης για την διαδικασία της εκπαίδευσης. Στο πεδίο της υπολογιστικής όρασης τα δεδομένα αυτά απαρτίζονται από εικόνες.

2.4 Σχετικές εργασίες

Η ανάπτυξη τεχνικών ανίχνευσης ζώων με σκοπό την προστασία κατά την οδήγηση δεν είναι ιδιαίτερα διαδεδομένη στην κοινότητα των προγραμματιστών που ασχολούνται με την υπολογιστική όραση. Αντιθέτως ιδιαίτερα διαδεδομένες ήταν η ανίχνευση πεζών και η ανίχνευση οδικών σημάτων. Επιπλέον πολλά από τα υπάρχοντα συστήματα ανίχνευσης ζώων δεν περιέχουν αναλυτικές περιγραφές σχετικά με την διαδικασία ανίχνευσης που εφαρμόζουν, όπως τα [14] και [15]. Αντιθέτως οι συγγραφείς του [16] και [17] έχουν αναλύσει λεπτομερώς τις μεθόδους ανίχνευσης ζώων που αναπτύξανε. Στο [16] γίνεται η χρήση θερμικής κάμερας για την λήψη εικόνων και των αλγορίθμων HOG και SVM για την κατηγοριοποίηση των δειγμάτων. Στο [17] οι εικόνες έχουν ληφθεί στο ορατό φάσμα και χρησιμοποιούν ένα σύστημα ανίχνευσης δύο σταδίων, κάνοντας χρήση των αλγορίθμων LBP [18]-Adaboost [19] και HOG-SVM για κάθε στάδιο αντίστοιχα.

Στο κομμάτι της ανίχνευσης ζώων με σκοπό την μελέτη στο [20] προτείνεται ένα σύστημα το οποίο τοποθετείται σε σημεία διέλευσης ζώων, καταγράφοντας έτσι την διέλευση ζώων και την μεταναστευτική τους ροή. Το σύστημα χρησιμοποιεί τους αλγόριθμους SVM, SIFT [21] και SURF [22].

Τέλος για την προστασία ζώων που φωλιάζουν στους αγρούς κατά την διάρκεια του θερισιμού, στο [23] χρησιμοποιούνται θερμικές κάμερες και αυτόνομα ελικόπτερα μικρού μεγέθους για τον αυτόματο εντοπισμό των ζώων.

Κεφάλαιο 3 Σχεδίαση και Υλοποίηση του Συστήματος

3.1 Ανάλυση της Αρχιτεκτονικής του Συστήματος

Η αρχιτεκτονική του συστήματος το οποίο υλοποιούμε στην παρούσα εργασία αποτελείται από έναν περιγραφέα και δύο στάδια ταξινόμησης της εικόνας που δίνεται σαν είσοδο. Ο στόχος αυτής της αρχιτεκτονικής είναι η βελτίωση της απόδοσης του συστήματος συγκριτικά με αρχιτεκτονικές ενός σταδίου, τόσο στο ποσοστό των επιτυχών ανιχνεύσεων όσο και στον χρόνο εκτέλεσης.

3.1.1 Ανάγκη ύπαρξης Περιγραφέα

Πιο αναλυτικά για να καταστήσουμε πιο αποδοτική την εκπαίδευση του αλγορίθμου SVM που θα χρησιμοποιήσουμε στο πρώτο στάδιο της ταξινόμησης των δειγμάτων, πρέπει να παρέχουμε την πληροφορία η οποία παρέχεται σαν είσοδος στον αλγόριθμο σε όσο το δυνατόν πιο ανθεκτική, απέναντι σε αλλοιώσεις, μορφή. Δεδομένου ότι η ανίχνευση των ζώων γίνεται σε εξωτερικούς χώρους όπου η φωτεινότητα μεταβάλλεται συνεχώς και ότι οι χρωματισμοί ζώων ίδιου είδους μεταβάλλονται ανάλογα με την εποχή, την ηλικία και την θερμοκρασία του περιβάλλοντος, η χρήση των εικόνων που χρησιμοποιούνται για την εκπαίδευση του πρώτου αλγορίθμου χωρίς περαιτέρω επεξεργασία δυσχεραίνει την κατηγοριοποίηση των δειγμάτων. Για αυτό το λόγο επιλέγουμε να χρησιμοποιήσουμε έναν περιγραφέα χαρακτηριστικών και πιο συγκεκριμένα τον αλγόριθμο HOG. Οι δύο λόγοι που επιλέξαμε τον συγκεκριμένο αλγόριθμο έναντι άλλων είναι:

- Η καλή συνεργασία του με τον ταξινομητή SVM [17].
- Κατάλληλο για χρήση σε αναζήτηση με ολισθαίνον παράθυρο.
- Η υλοποίηση του αλγορίθμου σε CUDA που περιείχε η βιβλιοθήκη OpenCV.

3.1.2 Ανάγκη ύπαρξης δύο σταδίων ταξινόμησης

Ο λόγος ύπαρξης δύο σταδίων ταξινόμησης είναι οι μειωμένες πιθανότητες που έχει ένα τέτοιο σύστημα να δώσει σαν έξοδο λανθάνον θετικό αποτέλεσμα. Εάν p_1 και p_2 οι πιθανότητες να χαρακτηριστεί ένα δείγμα σαν λανθάνον θετικό από το πρώτο και δεύτερο ταξινομητή αντίστοιχα τότε η πιθανότητα το ίδιο δείγμα να χαρακτηριστεί σαν λανθάνον θετικό και από τους δύο ταξινομητές είναι ίση με $p_1 \times p_2 < p_1, p_2$. Συνεπώς η χρήση δύο σταδίων ταξινόμησης επιφέρει μείωση του ποσοστού των λανθανόντων θετικών δειγμάτων με κόστος την αυξημένη υπολογιστική πολυπλοκότητα.

Η αύξηση της υπολογιστικής πολυπλοκότητας που προκύπτει είναι ένα ιδιαίτερα μεγάλο πρόβλημα που απαιτεί λύση. Συνεπώς αυτό που επιθυμούμε από τον ταξινομητή του πρώτου σταδίου είναι ο χρόνος που απαιτείται για την κατηγοριοποίηση ενός δείγματος να είναι σχετικά μικρός, απαραίτητα μικρότερος του δεύτερου ταξινομητή, και να βρίσκει όσο γίνεται περισσότερα αληθώς θετικά δείγματα κρατώντας το ποσοστό των λανθανόντων θετικών δειγμάτων σε ικανοποιητικά χαμηλά επίπεδα. Στον ρόλο αυτόν επιλέγουμε να χρησιμοποιήσουμε τον ταξινομητή γραμμικού SVM έναντι του CNN διότι θεωρούμε και αποδεικνύουμε πειραματικά, πως πληροί τις προαναφερθείσες προδιαγραφές.

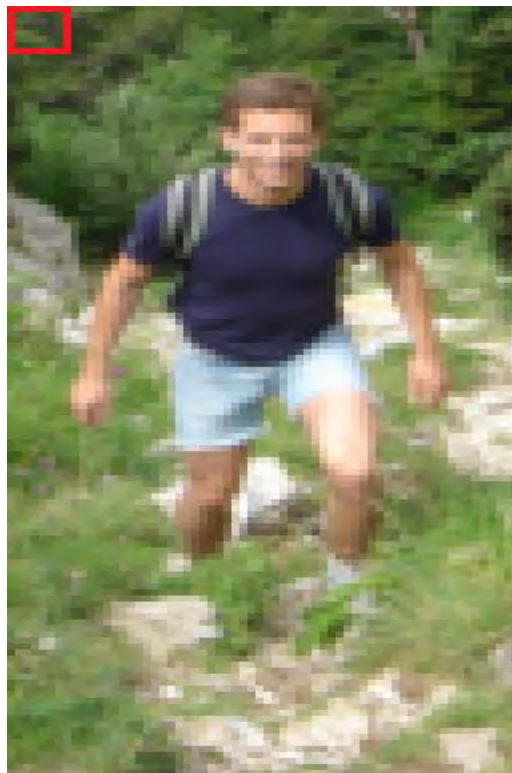
Έχοντας κατηγοριοποιήσει τα δείγματα μας στο πρώτο στάδιο προωθούμε τα δείγματα που χαρακτηρίστηκαν ως θετικά στον ταξινομητή του δεύτερου σταδίου. Τα δείγματα που θα κριθούν θετικά και στο δεύτερο στάδιο του συστήματος θα προκύψουν ως έξοδος από το σύστημα μας. Έχοντας χρησιμοποιήσει έναν γρήγορο στην εκτέλεση, αλλά όχι τόσο ισχυρό, ταξινομητή στο πρώτο στάδιο, θεωρούμε και αποδεικνύουμε πειραματικά, ότι απαιτείται η χρήση ενός πιο ισχυρού ταξινομητή σε δεύτερο στάδιο. Για αυτό τον σκοπό επιλέξαμε να χρησιμοποιήσουμε τα συνελκτικά νευρωνικά δίκτυα τα οποία θεωρούνται ως οι ταξινομητές με τους υψηλότερους δείκτες επιτυχίας αυτή τη στιγμή.

3.2 Περιγραφέας HOG

Ο αλγόριθμος HOG(Histogram of Oriented Gradients) είναι ένας περιγραφέας χαρακτηριστικών. Σκοπός του είναι, όπως και ο σκοπός όλων των περιγραφών χαρακτηριστικών, η γενικευμένη περιγραφή ενός εικονιζόμενου αντικειμένου, με τέτοιο τρόπο, ώστε η περιγραφή του να είναι η ίδια ακόμα και εάν το συγκεκριμένο αντικείμενο εμφανίζεται υπό διαφορετικές συνθήκες. Αυτό κάνει το έργο του ταξινομητή ευκολότερο. Ο συγκεκριμένος περιγραφέας χρησιμοποιείται συχνά μαζί με το ταξινομητή SVM σε εφαρμογές αναγνώρισης πεζών, με μεγάλη επιτυχία.

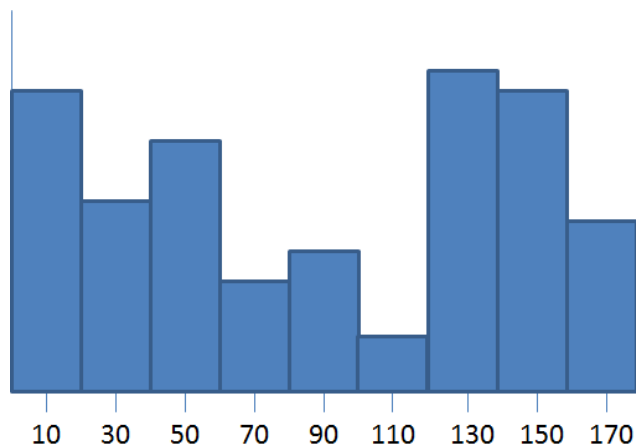
Ο αλγόριθμος HOG είναι απλός και σχετικά εύκολος στην κατανόηση. Βασική ιδιότητα του είναι η χρήση ενός γενικού χαρακτηριστικού για την περιγραφή ενός αντικειμένου, έναντι πολλών τοπικών χαρακτηριστικών που χρησιμοποιούν άλλοι αλγόριθμοι. Αυτό σημαίνει ότι ολόκληρο το αντικείμενο αναπαρίσταται από ένα χαρακτηριστικό σε μορφή διανύσματος, αντί πολλών χαρακτηριστικών διανυσμάτων που αναπαριστούν μικρότερες περιοχές του αντικειμένου. Δεν προκύπτει κάποιο πλεονέκτημα ή μειονέκτημα από αυτή του την ιδιότητα καθώς η λειτουργία του συγκεκριμένου αλγορίθμου είναι η μόνη συμβατή προς τις λειτουργίες του συστήματος μας(αναζήτηση ολισθαίνου παραθύρου).

Ο περιγραφέας HOG παίρνει σαν είσοδο μια εικόνα και υπολογίζει ένα χαρακτηριστικό το οποίο αφορά την πληροφορία που περιέχει. Το μέγεθος της εισόδου είναι μεταβλητό και ορίζεται ανάλογα με τις ανάγκες της κάθε υλοποίησης. Το μέγεθος που επιλέξαμε εμείς είναι 32x32, ίδιο με το μέγεθος των εικόνων που περιλαμβάνονται στο σύνολο δεδομένων που χρησιμοποιήσαμε.



Εικόνα 3.1 Παράδειγμα ολισθαίνωντος παραθύρου.
[28]

Για τον υπολογισμό του χαρακτηριστικού διανύσματος ο περιγραφέας ακολουθεί κάποια συγκεκριμένα βήματα. Το πρώτο βήμα είναι ο υπολογισμός του ιστογράμματος κλίσεων. Για αυτό τον υπολογισμό χωρίζουμε το παράθυρο σε τετράγωνα ίσου μεγέθους τα οποία αποκαλούμε κελιά. Στην υλοποίηση μας τα κελιά (*cells*) έχουν μέγεθος 8×8 . Μέσα στα κελιά υπολογίζουμε το διάνυσμα κλίσεων (*gradient vector*) για κάθε ένα από τα εικονοστοιχεία που περιλαμβάνονται. Έτσι παίρνουμε 64 διανύσματα κλίσεων τα οποία κατανέμουμε σε ένα ιστόγραμμα 9 θέσεων ανάλογα με την γωνία της



Εικόνα 3.2 Ιστόγραμμα κλίσεων [28]

κατεύθυνσης της κλίσης που υπάρχει στο συγκεκριμένο εικονοστοιχείο. Το ιστόγραμμα περιλαμβάνει τις γωνίες από 0 έως 180, δηλαδή κάθε θέση του αντιστοιχεί σε 20 μοίρες.

Για τον υπολογισμό του ιστογράμματος κλίσεων κάθε εικονοστοιχείο έχει διαφορετική συνεισφορά ανάλογη του μεγέθους της απόλυτης τιμής του διανύσματος. Η κάθε συνεισφορά χωρίζεται στις δύο κοντινότερες θέσεις ανάλογα με την κατεύθυνση της κλίσης. Για παράδειγμα εάν ένα διάνυσμα έχει κατεύθυνση 85 μοίρες, το $\frac{1}{4}$ της συνεισφοράς του θα πάει στην θέση που αντιστοιχεί στις 70 μοίρες και τα υπόλοιπα $\frac{3}{4}$ στην θέση που αντιστοιχεί στις 90 μοίρες.

Το επόμενο βήμα είναι η κανονικοποίηση των ιστογραμμάτων. Για αυτή την διαδικασία ομαδοποιούμε τα κελιά σε μεγαλύτερα σύνολα με την ονομασία μπλόκ (*blocks*) και κάνουμε την κανονικοποίηση σύμφωνα με τα κελιά που υπάρχουν μέσα στο μπλόκ. Στην υλοποίηση μας τα κελιά έχουν μέγεθος 16x16 και αλληλοεπικαλύπτονται κατά 50%. Η κανονικοποίηση γίνεται ενώνοντας τα ιστόγραμμα κάθε μπλόκ σε ένα μεγάλο διάνυσμα μεγέθους 36 θέσεων και διαιρώντας τα με την απόλυτη τιμή του διανύσματος.

Τέλος τα κανονικοποιημένα διανύσματα του κάθε μπλόκ συγκεντρώνονται σε ένα διάνυσμα μεγέθους:



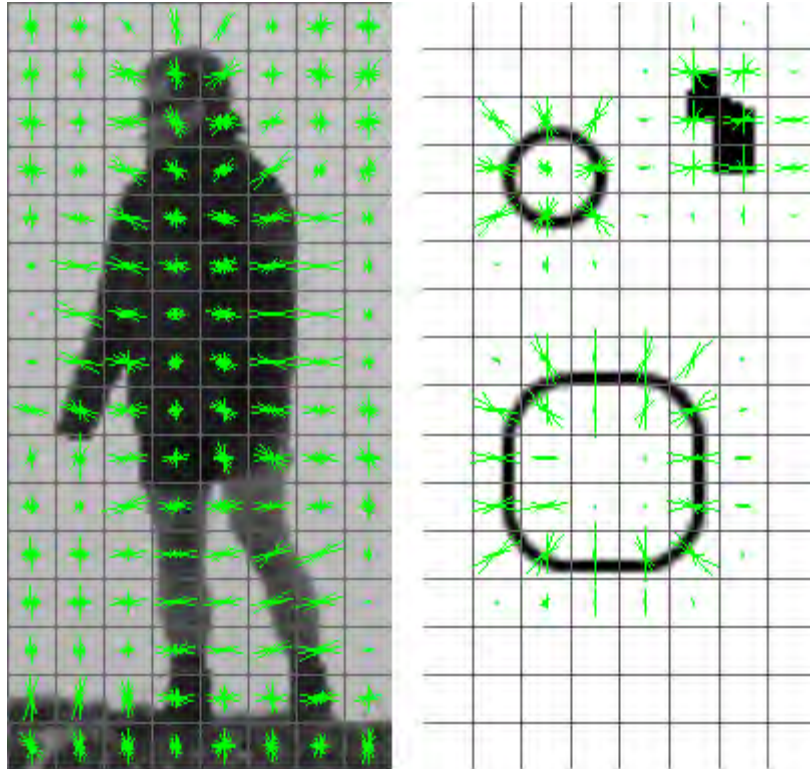
Εικόνα 3.3 Αναπαράσταση των μπλοκ και της αλληλοεπικάλυψής τους. [28]

$$\text{μέγεθος διανύσματος} = \# \text{ μπλοκ} \times \# \text{ κελιών ανά μπλοκ} \times \# \text{ θέσεων ανά ιστόγραμμα}$$

Το διάνυσμα αυτό αποτελεί το χαρακτηριστικό της εικόνας.

Στην υλοποίηση μας το χαρακτηριστικό-διάνυσμα είναι μεγέθους 324 θέσεων. Αυτό σημαίνει πως την αρχική εικόνα 32x32 συνολικού μεγέθους 1024 θέσεων (ασπρόμαυρη) που θα δίναμε στον

ταξινομητή σαν είσοδο, την αντικαθιστούμε με ένα χαρακτηριστικό μεγέθους 324 θέσεων. Έτσι μειώνουμε τον αριθμό των πράξεων που απαιτούνται για την ταξινόμηση ενός δείγματος. Ένα επιπλέον κέρδος που έχουμε πέραν της μείωσης του μεγέθους της εισόδου είναι η μεγαλύτερη ανθεκτικότητα των δειγμάτων στις αλλαγές.



Εικόνα 3.4 Παραδείγματα εφαρμογής του περιγραφέα HOG. [28]

Για την υλοποίηση του περιγραφέα HOG χρησιμοποιήθηκε η βιβλιοθήκη OpenCV 3.1. Τα μεγέθη που χρησιμοποιήθηκαν για τα κελιά, τα μπλόκ και τις θέσεις του ιστογράμματος επιλέχθηκαν διότι είναι τα μόνα που υποστηρίζονται από την OpenCV για τον υπολογισμό του διανύσματος περιγραφής κάνοντας χρήση της GPU.

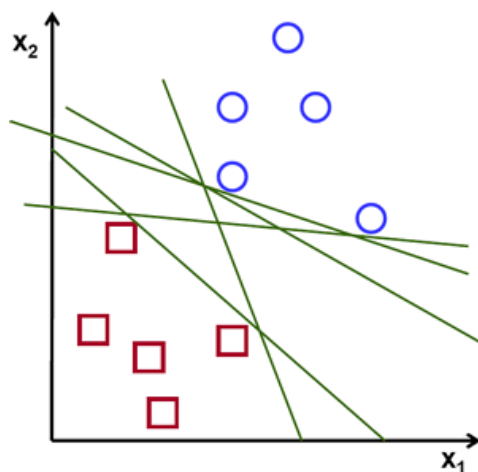
3.3 Ταξινόμηση με χρήση SVM

Μετά την δημιουργία των χαρακτηριστικών δίνουμε σαν είσοδο στον ταξινομητή το δείγμα μας με σκοπό να λάβουμε σαν έξοδο την κατηγορία στην οποία θεωρεί ο ταξινομητής ότι ανήκει. Πριν γίνει αυτό έχουμε φροντίσει να εφαρμόσουμε την διαδικασία της εκπαίδευσης στο μοντέλο του ταξινομητή και είμαστε ικανοποιημένοι με τους δείκτες της αποδοτικότητας του.

3.3.1 Ανάλυση αλγορίθμου SVM

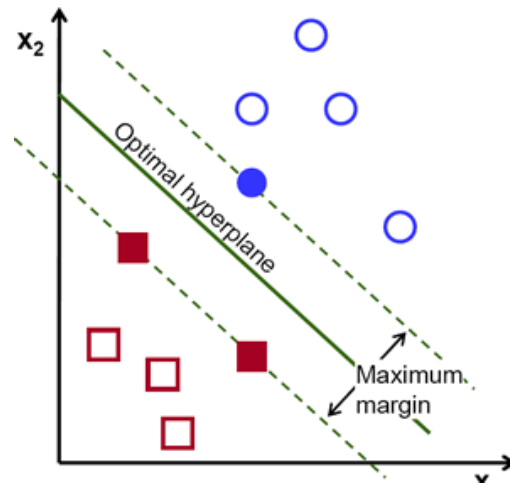
Ο αλγόριθμος SVM είναι ένας ταξινομητής επιτηρούμενης μάθησης (*supervised learning*) ο οποίος ορίζεται από ένα διαχωριστικό υπερεπίπεδο. Με άλλα λόγια, δοθέντος ενός συνόλου δεδομένων του οποίου οι κατηγορίες των δειγμάτων είναι ήδη γνωστές, ο αλγόριθμος επιστρέφει το βέλτιστο υπερεπίπεδο το οποίο ταξινομεί επιτυχώς δείγματα που θα δοθούν σαν είσοδο στο μέλλον.

Υπάρχουν πολλά υπερεπίπεδα που να μπορούν να κατηγοριοποιήσουν τα δείγματα που περιέχονται στο σύνολο δεδομένων που χρησιμοποιούμε για την εκπαίδευση του ταξινομητή.



Εικόνα 3.5 Παράδειγμα διαχωριστικών υπερεπιπέδων στον δισδιάστατο χώρο. [29]

Όμως το βέλτιστο διαχωριστικό υπερεπίπεδο ξεχωρίζει διότι έχει την μεγαλύτερη ελάχιστη απόσταση από τα δείγματα εκπαίδευσης. Έτσι το συγκεκριμένο υπερεπίπεδο μεγιστοποιεί το περιθώριο (*margin*) με τα δείγματα εκπαίδευσης.



Εικόνα 3.6 Βέλτιστο υπερεπίπεδο [29]

Ένα υπερεπίπεδο ορίζεται από την συνάρτηση:

$$f(x) = \beta_0 + \beta^T x$$

όπου β είναι το διάνυσμα βαρών και το β_0 είναι η πόλωση.

Το βέλτιστο υπερεπίπεδο μπορεί να αναπαρασταθεί με άπειρους τρόπους, απλά κλιμακώνοντας τα β και β_0 . Για θέματα σύμβασης από όλες τις δυνατές αναπαραστάσεις του υπερεπίπεδου αυτή που επιλέγεται είναι:

$$|\beta_0 + \beta^T x| = 1$$

όπου το x συμβολίζει τα δείγματα εκπαίδευσης που βρίσκονται πιο κοντά στο υπερεπίπεδο. Συνήθως τα πιο κοντινά δείγματα εκπαίδευσης στον υπερεπίπεδο λέγονται διανύσματα υποστήριξης (*support vectors*). Αυτή η αναπαράσταση είναι γνωστή ως κανονικό υπερεπίπεδο.

Τώρα χρησιμοποιώντας γεωμετρία βρίσκουμε την απόσταση ενός σημείου x και ενός υπερεπίπεδου (β, β_0) :

$$\text{απόσταση} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|}$$

Συγκεκριμένα για το κανονικό υπερεπίπεδο, ο αριθμητής είναι ίσος με ένα οπότε η απόσταση προς τα διανύσματα υποστήριξης είναι:

$$\text{αποσταση}_{\text{διανύσματα υποστήριξης}} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|}$$

Συνεπώς το μέγιστο περιθώριο το ισούται με το διπλάσιο της απόστασης στο κοντινότερο δείγμα οπότε είναι ίσο με:

$$M = \frac{2}{\|\beta\|}$$

Τέλος το πρόβλημα της μεγιστοποίησης του περιθωρίου M ανάγεται στο πρόβλημα γραμμικού προγραμματισμού το οποίο καλείται να ελαχιστοποιήσει την συνάρτηση $L(\beta)$ η οποία υπόκειται σε περιορισμούς. Οι περιορισμοί αναπαριστούν την απαίτηση της ορθής κατηγοριοποίησης των εκπαιδευτικών δειγμάτων x_i . Επισήμως το πρόβλημα εκφράζεται ως:

$$\min_{\beta, \beta_0} L(\beta) = \frac{1}{2} \|\beta\|^2 \text{ με τους περιορισμούς } y_i(\beta^T x_i + \beta_0) \geq 1 \forall i$$

όπου το y_i αναπαριστά τις κατηγορίες των δειγμάτων εκπαίδευσης.

3.3.2 Υλοποίηση του αλγορίθμου SVM

Η υλοποίηση που χρησιμοποιήθηκε για το σύστημα που μελετούμε, του ταξινομητή γραμμικού SVM, περιλαμβάνεται στην βιβλιοθήκη της OpenCV. Για την εκπαίδευση του αλγορίθμου χρησιμοποιήθηκε το αντίστοιχο εργαλείο που περιέχεται στην OpenCV και το οποίο είναι παρακλάδι της βιβλιοθήκης LIBSVM [24] και πλέον αναπτύσσεται ξεχωριστά. Από την εκπαίδευση του αλγορίθμου προκύπτουν 6707 διανύσματα υποστήριξης. Για την κατηγοριοποίηση νέων δειγμάτων χρησιμοποιήθηκε η υλοποίηση του αλγορίθμου SVM σε CUDA που επίσης περιέχεται στην βιβλιοθήκη OpenCV. Για την εκπαίδευση του αλγορίθμου χρησιμοποιήθηκε το σύνολο δεδομένων CIFAR-10 το οποίο περιείχε και επιπλέον εικόνες από φυσικά τοπία για την καλύτερη απόδοση του σε εικόνες όπου τα ζώα βρίσκονται στο φυσικό τους περιβάλλον. Τα πειραματικά αποτελέσματα αναλύονται στο τέταρτο κεφάλαιο.

3.4 Ταξινόμηση με χρήση CNN

3.4.1 Νευρωνικά Δίκτυα

Στον χώρο της μηχανικής εκμάθησης τα τεχνητά νευρωνικά (*Artificial Neural Networks*) δίκτυα είναι μια οικογένεια μοντέλων, εμπνευσμένα από τα βιολογικά νευρωνικά δίκτυα (*Biological Neural Networks*). Χρησιμοποιούνται για την εκτίμηση και την προσέγγιση συναρτήσεων οι οποίες χρησιμοποιούν σαν είσοδο ένα μεγάλο αριθμό μεταβλητών. Τα νευρωνικά δίκτυα μοντελοποιούνται ως συστήματα διασυνδεδεμένων νευρώνων τα οποία ανταλλάσσουν μηνύματα μεταξύ τους. Οι συνδέσεις μεταξύ των νευρώνων διαθέτουν βάρη τα οποία μπορούν να μεταβληθούν ανάλογα με την εμπειρία, καθιστώντας τα νευρωνικά δίκτυα ικανά να προσαρμόζονται σε διαφορετικές εισόδους και να μαθαίνουν.

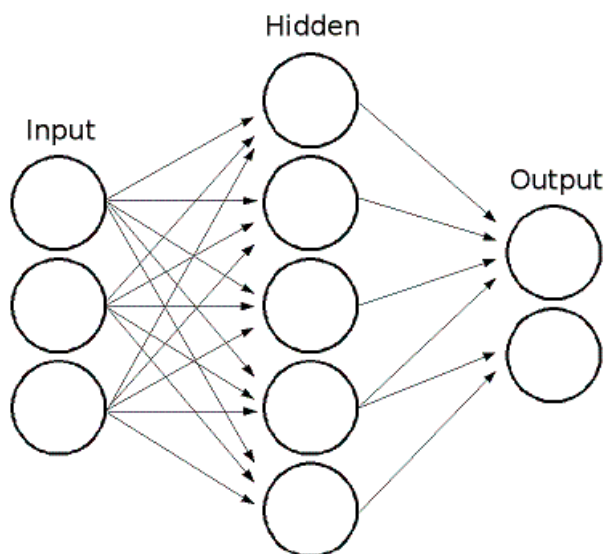
Στην επιτηρούμενη μάθηση διαθέτουμε ένα σύνολο δειγμάτων της μορφής (x, y) , $x \in X$, $y \in Y$ και ο στόχος είναι η εύρεση μιας συνάρτησης $f: X \rightarrow Y$ η οποία να μπορεί να αντιστοιχίσει τα δείγματα. Με άλλα λόγια επιθυμούμε να συνάγουμε την αντιστοιχίση που συνάγεται από τα δεδομένα. Η συνάρτηση κόστους μετράει την απόσταση της αντιστοιχίσης και τον δεδομένων που παρέχονται.

Μια συχνώς χρησιμοποιούμενη συνάρτηση κόστους είναι το ελάχιστο τετραγωνικό σφάλμα το οποίο προσπαθεί να ελαχιστοποιήσει το μέσο τετραγωνικό σφάλμα μεταξύ της εξόδου του νευρωνικού δικτύου, $f(x)$, και της κατηγορίας y όπου ανήκει το x . Όταν χρησιμοποιούμε την μέθοδο της κλίσης

καθόδου (*gradient descent*) για νευρωνικά δίκτυα της κατηγορίας νευρώνων Perceptron, τότε προκύπτει ο αλγόριθμος της οπισθοδιάδοσης λάθους που χρησιμοποιείται για την εκπαίδευση των νευρωνικών δικτύων.

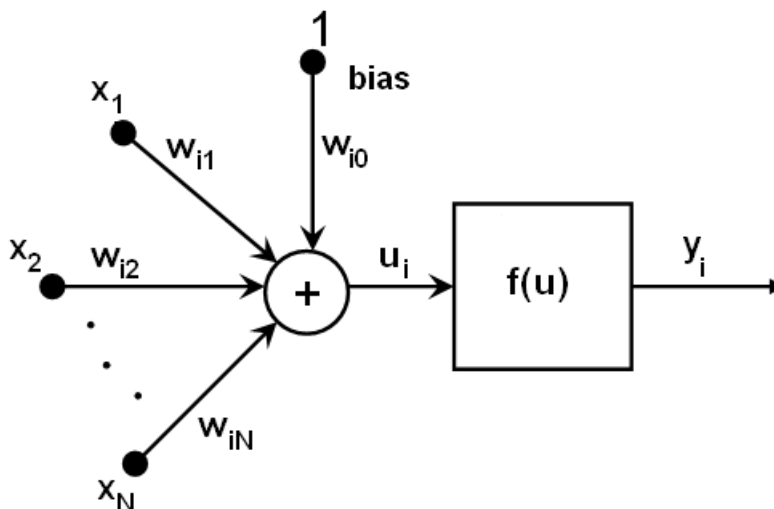
Ο αλγόριθμος της οπισθοδιάδοσης λάθους υπολογίζει την κλίση της συνάρτησης κόστους λαμβάνοντας υπόψη τα βάρη των νευρώνων του δικτύου. Η κλίση δίνεται σαν είσοδος στην μέθοδο βελτιστοποίησης η οποία με την σειρά της υπολογίζει τα νέα βάρη των νευρώνων, με σκοπό την ελαχιστοποίηση της συνάρτησης κόστους.

Τα πιο συχνά χρησιμοποιούμενα νευρωνικά δίκτυα είναι οι πολυεπίπεδοι νευρώνες Perceptron (*Multi-Layer Perceptron*). Τα MLP αποτελούνται από ένα επίπεδο εισόδου, ένα επίπεδο εξόδου και ένα ή περισσότερα κρυφά επίπεδα.



Εικόνα 3.7 Παράδειγμα MLP 3 επιπέδων, με 3 εισόδους, 2 εξόδους και 1 κρυφό επίπεδο 5 νευρώνων. [31]

Όλοι οι νευρώνες των MLP είναι παρόμοιοι. Κάθε ένας από αυτούς διαθέτει πολλούς νευρώνες εισόδου και πολλούς νευρώνες εξόδου. Οι τιμές που προέρχονται από τα προηγούμενα επίπεδα αθροίζονται, ανάλογα με τα βάρη, ανεξάρτητα για κάθε νευρώνα και σε αυτό το άθροισμα προστίθεται ο όρος της πόλωσης. Το άθροισμα μετασχηματίζεται χρησιμοποιώντας την συνάρτηση ενεργοποίησης f η οποία μπορεί να διαφοροποιείται από νευρώνα σε νευρώνα.



Εικόνα 3.8 Παράδειγμα άθροισης των εισόδων στον νευρώνα [31]

Με άλλα λόγια, δεδομένων των εξόδων x_j του επιπέδου n , οι έξοδοι y_i του επιπέδου $n + 1$ υπολογίζονται ως:

$$u_i = \sum_j (w_{i,j}^{n+1} * x_j) + w_{i,bias}^{n+1}$$

$$y_i = f(u_i)$$

Για την συνάρτηση ενεργοποίησης f μπορούν να χρησιμοποιηθούν διαφόρων ειδών συναρτήσεις όπως:

- Η ταυτοτική συνάρτηση $f(x) = x$
- Η σιγμοϊδής συνάρτηση $f(x) = \frac{\beta * (1 - e^{-ax})}{(1 + e^{-ax})}$ όπου συνήθως $\alpha = 1, \beta = 1$
- Η γκαουσιανή συνάρτηση $f(x) = \beta e^{-ax * x}$

Συνοψίζοντας ένα νευρωνικό δίκτυο λειτουργεί ως εξής:

1. Δίνουμε την εικόνα ή το χαρακτηριστικό που θέλουμε με την μορφή διανύσματος. Το επίπεδο εισόδου πρέπει να έχει ίδιο μέγεθος με το διάνυσμα .
2. Περνάμε τις τιμές του επιπέδου εισόδου στο πρώτο κρυφό επίπεδο.
3. Υπολογίζουμε την έξοδο του κάθε κρυφού επιπέδου χρησιμοποιώντας τα βάρη και τις συναρτήσεις ενεργοποίησης .
4. Περνάμε τις εξόδους στα χαμηλότερα επίπεδα και επαναλαμβάνουμε μέχρι να φτάσουμε στο επίπεδο εξόδου.

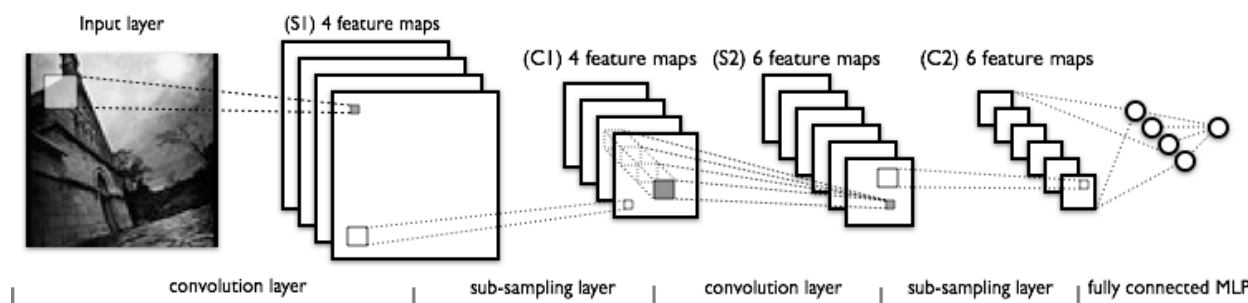
3.4.2 Συνελικτικά Νευρωνικά Δίκτυα

Τα συνελικτικά νευρωνικά δίκτυα ανήκουν σε ένα παρακλάδι του τομέα της μηχανικής εκμάθησης που ονομάζεται βαθιά εκμάθηση (Deep Learning) [25]. Οι αλγόριθμοι που ανήκουν σε αυτόν τον τομέα επιχειρούν να μοντελοποιήσουν υψηλού επιπέδου αφαιρέσεις σε δεδομένα, χρησιμοποιώντας πολλαπλά επίπεδα επεξεργασίας, με περίπλοκες δομές αποτελούμενες από πολλαπλούς μη γραμμικούς μετασχηματισμούς.

Τα συνελικτικά νευρωνικά δίκτυα (*Convolutional Neural Networks - CNN*) είναι ένα είδος τεχνητών νευρωνικών δικτύων των οποίων ο τρόπος διασύνδεσης των νευρώνων είναι οργανωμένος όπως ο οπτικός φλοιός των ζώων, όπου ανεξάρτητοι νευρώνες είναι τοποθετημένοι σε τέτοια διάταξη ώστε να αντιστοιχούν σε επικαλυπτόμενες περιοχές του οπτικού πεδίου. Ο σκοπός των συνελικτικών νευρωνικών δικτύων είναι η μείωση της ανάγκης για προεπεξεργασία των δεδομένων.

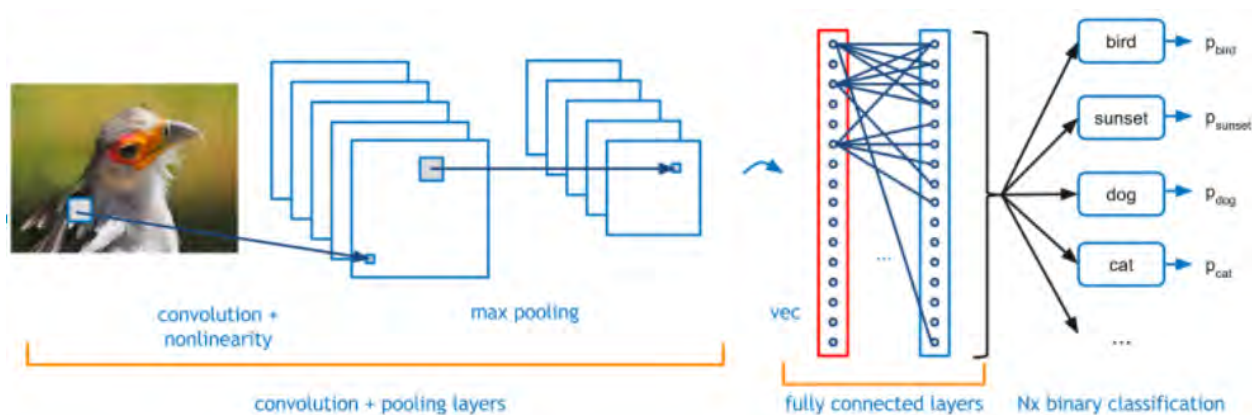
Τα επίπεδα τα οποία περιέχονται σε ένα συνελικτικό νευρωνικό δίκτυο έχουν ξεχωριστές λειτουργίες και ιδιότητες. Επίσης μπορούν να χρησιμοποιηθούν επανειλημμένα μέσα στο ίδιο δίκτυο για την βελτίωση των αποτελεσμάτων. Παραδείγματα τέτοιων επιπέδων είναι τα ακόλουθα:

- Επίπεδο συνέλιξης: Πραγματοποιεί συνέλιξη της εικόνας εισόδου με ένα σετ από φίλτρα καθένα από τα οποία παράγει ένα χαρακτηριστικό για κάθε σημείο της εικόνας. Χαρακτηριστικό είναι πως αυτά τα φίλτρα δεν είναι όμοια μεταξύ τους και μεταβάλλονται ανάλογα με τις ανάγκες του μοντέλου. Οι τιμές των φίλτρων προσδιορίζονται κατά την διαδικασία της εκπαίδευσης του μοντέλου.
- Επίπεδο υποδειγματοληψίας/συγχώνευσης: Σκοπός του είναι η μείωση του μεγέθους του σήματος με το να αντιστοιχίζει μια περιοχή του σήματος σε μια θέση. Η τιμή που θα πάρει αυτή ή θέσει εξαρτάται από την υλοποίηση. Μπορεί να λαμβάνει μια τυχαία τιμή από τις υπάρχουσες στην περιοχή ή μπορεί να επιλέγει την μεγαλύτερη.
- Εξομάλυνση τοπικής απάντησης (Local Response Normalization) : πραγματοποιεί ένα είδος πλευρικής αναστολής (Lateral Inhibition) κανονικοποιώντας τοπικές περιοχές εισόδου.



Εικόνα 3.9 Αναπαράσταση του συνελικτικού νευρωνικού δικτύου LeNet. [30]

Αξίζει να σημειωθεί ότι ένα συνελικτικό νευρωνικό δίκτυο μπορεί να αντιπροσωπεύσει τα πρώτα επίπεδα ενός τεχνητού νευρωνικού δικτύου αποφεύγοντας έτσι την ανάγκη για προεπεξεργασία.



Εικόνα 3.10 Αναπαράσταση συνελκτικού νευρωνικού δικτύου μαζί με τεχνητό νευρωνικό δίκτυο [30]

3.4.3 Υλοποίηση του ταξινομητή CNN

Εφόσον το συνελκτικό νευρωνικό δίκτυο καθιστά περιττή την προεπεξεργασία της εισόδου, στο δεύτερο επίπεδο ταξινόμησης δεν απαιτείται η χρήση κάποιου περιγραφέα. Στην υλοποίηση μας επιλέγουμε να χρησιμοποιήσουμε το συνελκτικό νευρωνικό δίκτυο CIFAR-10 το οποίο περιλαμβάνει η βιβλιοθήκη βαθείας εκμάθησης Caffe. Το συγκεκριμένο δίκτυο αποτελείται από 10 επίπεδα, το μέγεθος εισόδου είναι 3072 σημεία και έχει 10 εξόδους. Στην υλοποίηση μας η κατηγοριοποίηση που εκτελούμε είναι δυαδική οπότε τροποποιήσαμε τον αριθμό των εξόδων του δικτύου από 10 σε 2.

Η είσοδος του δικτύου είναι μια έγχρωμη εικόνα διαστάσεων 32x32. Οπότε βασική προϋπόθεση για την ορθή λειτουργία του ταξινομητή είναι η μετατροπή όλων των εικόνων εισόδου σε έγχρωμες εικόνες διαστάσεων 32x32. Οι διαδικασίες της ταξινόμησης και της εκπαίδευσης κάνουν χρήση της GPU με σκοπό την μείωση του χρόνου που απαιτείται για την ολοκλήρωσή τους. Για την διαδικασία της εκπαίδευσης χρησιμοποιήθηκε το ίδιο σύνολο δεδομένων που χρησιμοποιήθηκε και στον ταξινομητή πρώτου σταδίου.

Κεφάλαιο 4 Αποτελέσματα

4.1 Εκπαίδευση αλγορίθμων

Για την σωστή λειτουργία των ταξινομητών πρέπει πρώτα να εκπαιδεύσουμε τους αλγορίθμους. Στο σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση, το CIFAR-10, έχουμε ενσωματώσει εικόνες φυσικών τοπίων. Η προσθήκη αυτών των εικόνων που περιέχουν φυσικά τοπία έγινε με σκοπό την βελτίωση των αποτελεσμάτων όταν δίνουμε σαν είσοδο, εικόνες ζώων το οποία βρίσκονται σε φυσικά τοπία και πρέπει να γίνει κλιμακωτή ανίχνευση σε όλη την εικόνα.

Το σύνολο δεδομένων CIFAR-10 περιέχει 10 κλάσεις αντικειμένων:

- Αεροπλάνα
- Πουλιά
- Γάτες
- Ελάφια
- Σκύλους
- Βατράχια
- Άλογα
- Πλοία
- Οχήματα

Επίσης, περιέχει 50000 εικόνες που προορίζονται για την εκπαίδευση των αλγορίθμων, δηλαδή 5000 για κάθε κλάση. Τέλος, διαθέτει 10000 εικόνες για τον έλεγχο της διαδικασίας εκπαίδευσης.

4.1.1 Εκπαίδευση SVM

Για την εκπαίδευση του αλγορίθμου SVM ήταν αναγκαία η περαιτέρω τροποποίηση του συνόλου δεδομένων. Επειδή ο σκοπός του συστήματος είναι να αναγνωρίζει ένα συγκεκριμένο ζώο θέσαμε σαν θετική την κλάση του ζώου που θέλαμε να αναγνωρίσουμε και δώσαμε στην αρνητική κλάση συγκεντρώσαμε όλες τις υπόλοιπες κλάσεις. Αυτό έχει σαν αποτέλεσμα να δημιουργηθούν δύο κλάσεις, η θετική με 5000 δείγματα και η αρνητική με 45000 δείγματα. Αυτή η ανισορροπία στο μέγεθος των κλάσεων προκάλεσε προβλήματα κατά την εκπαίδευση μειώνοντας δραματικά το πλήθος των αληθώς θετικών δειγμάτων που ανιχνεύονταν από τον ταξινομητή. Η λύση σε αυτό το πρόβλημα δόθηκε με την μείωση των εικόνων που υπήρχαν στην αρνητική κλάση παίρνοντας μόνο 555 δείγματα από κάθε κλάση αντί των 5000 που λαμβάναμε αρχικά. Επίσης οι κλάσεις πλοίων και αεροπλάνων αφαιρέθηκαν εξ ολοκλήρου και στην θέση τους προστέθηκαν 1000 εικόνες από τοπία.

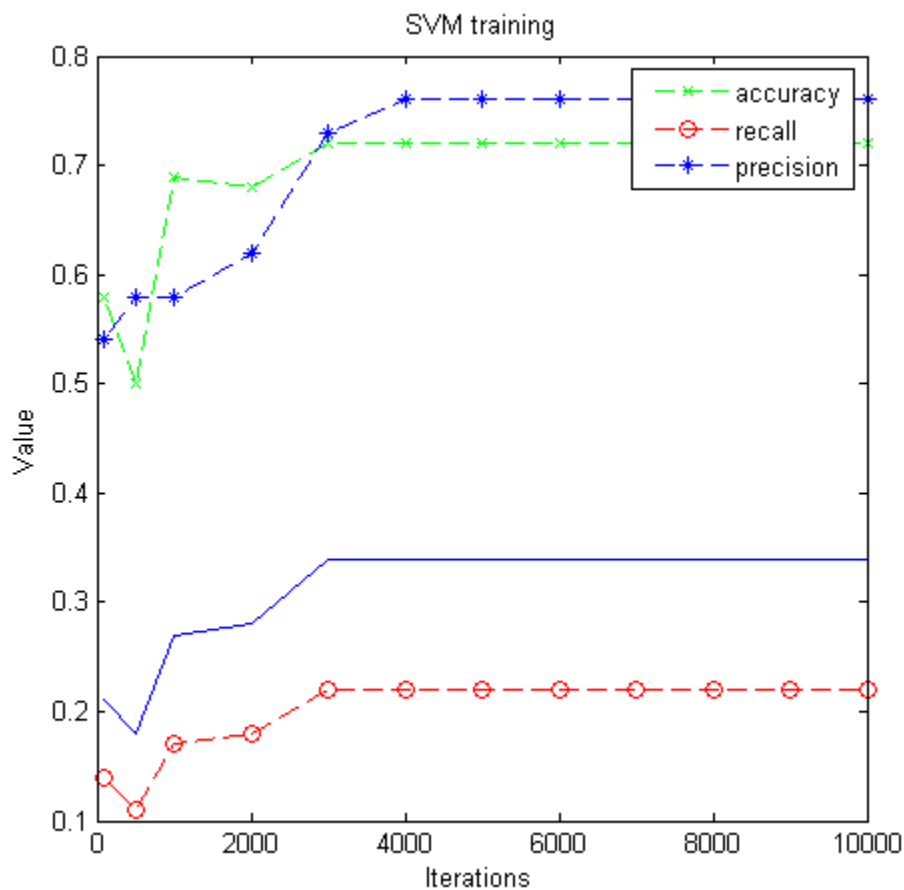
Τέλος εκτελώντας την διαδικασία την εκπαίδευσης 6000 φορές παίρνουμε μοντέλο που θα χρησιμοποιήσει ο ταξινομητής μας. Ο συγκεκριμένος αριθμός επαναλήψεων επιλέχθηκε διότι παρατηρήσαμε ότι μετά από έναν συγκεκριμένο αριθμό επαναλήψεων οι μετρικές δεν βελτιώνονται(εικόνα 4.1). Για να μετρήσουμε την απόδοση ενός μοντέλου χρησιμοποιούμε τις παρακάτω μετρικές:

- Ευστοχία (*accuracy*): ο λόγος των ορθών ανιχνεύσεων προς τα συνολικά δείγματα.
- Ακρίβεια (*precision*): η λόγος των ορθώς θετικών ανιχνεύσεων προς το σύνολο των θετικών ανιχνεύσεων.
- Ανάκληση (*recall*): η λόγος των ορθώς θετικών ανιχνεύσεων προς το συνολικά θετικά δείγματα.
- F1-σκορ(*f1-score*): η αρμονική μέση τιμή της ανίχνευσης και της ανάκλησης

Οι τιμές των μετρικών που δίνει το μοντέλο για το σύνολο δεδομένων ελέγχου είναι:

- Ευστοχία (*accuracy*) = 0.7097
- Ακρίβεια (*precision*) = 0.2208
- Ανάκληση (*recall*) = 0.753
- F1-σκορ(*f1-score*) = 0.3417

Αν και η ακρίβεια είναι χαμηλή, η ανάκληση και η ευστοχία είναι σε ικανοποιητικά επίπεδα. Άλλωστε αυτές ήταν και οι απαιτήσεις από το πρώτο στάδιο ταξινόμησης, όσο γίνεται μεγαλύτερη ανάκληση διατηρώντας την ευστοχία σε υψηλά επίπεδα.



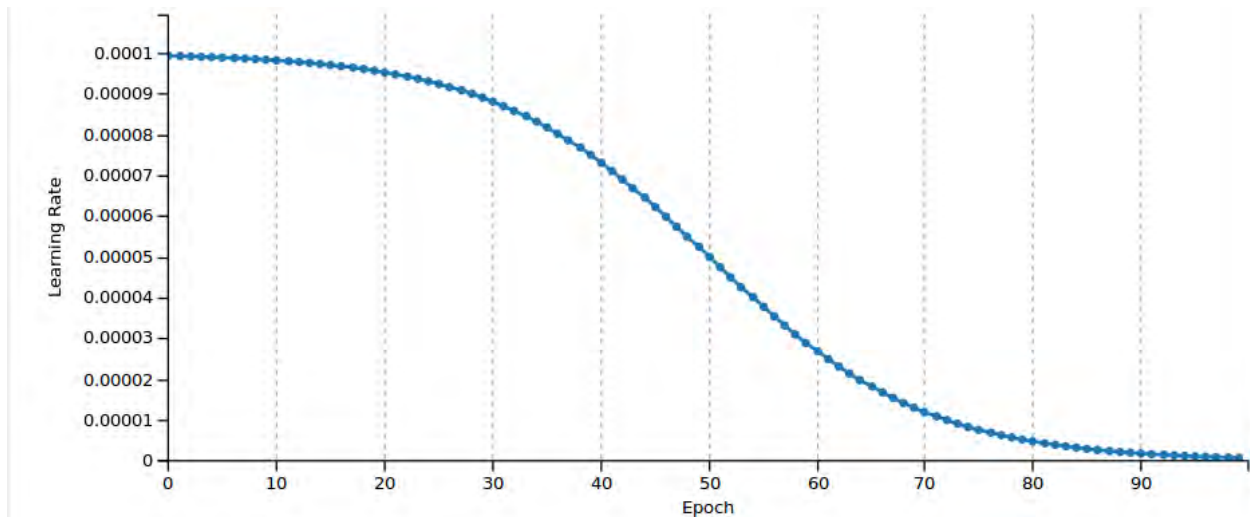
Εικόνα 4.1 Μετρικές κατά την εκπαίδευση του γραμμικού SVM

4.1.2 Εκπαίδευση CNN

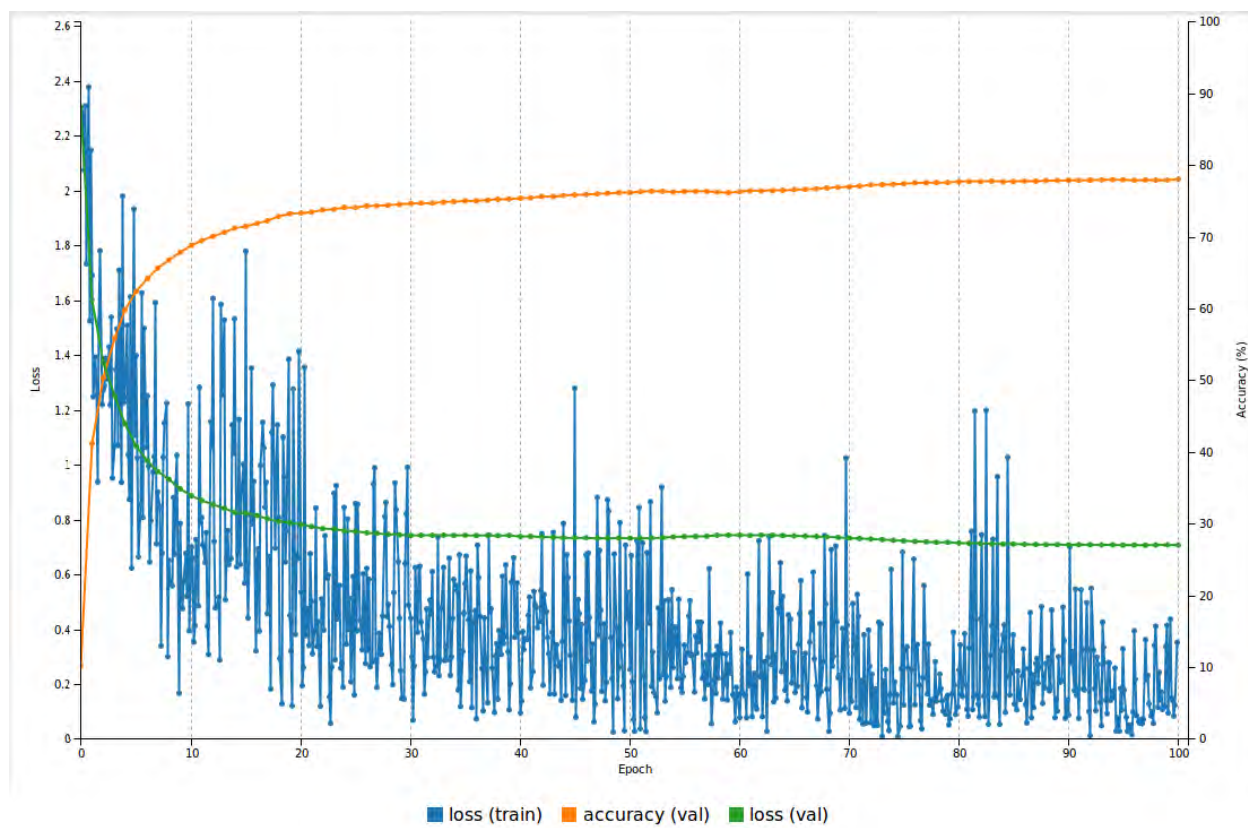
Για την εκπαίδευση του αλγορίθμου CNN η μόνη τροποποίηση που κάναμε στο σύνολο δεδομένων ήταν η συγχώνευση των εικόνων φυσικών τοπίων. Δεν υπήρξαν προβλήματα κατά την εκπαίδευση που να σχετίζονταν με τη διαφορά του μεγέθους των κλάσεων.

Η διαδικασία της εκπαίδευσης επαναλήφθηκε 60000 φορές. Ο αριθμός των επαναλήψεων επιλέχθηκε για τον ίδιο λόγο που αναφέραμε στον ταξινομητή SVM. Οι μετρικές που δίνει το μοντέλο για το σύνολο δεδομένων ελέγχου είναι:

- Ευστοχία (*accuracy*) = 0.7532
- Ακρίβεια (*precision*) = 0.2782
- Ανάκληση (*recall*) = 0.9210
- F1-σκορ (*f1-score*) = 0.4274



Εικόνα 4.2 Σιγμοϊδής συνάρτηση μεταβολής του ρυθμού μάθησης κατά την εκπαίδευση του CNN



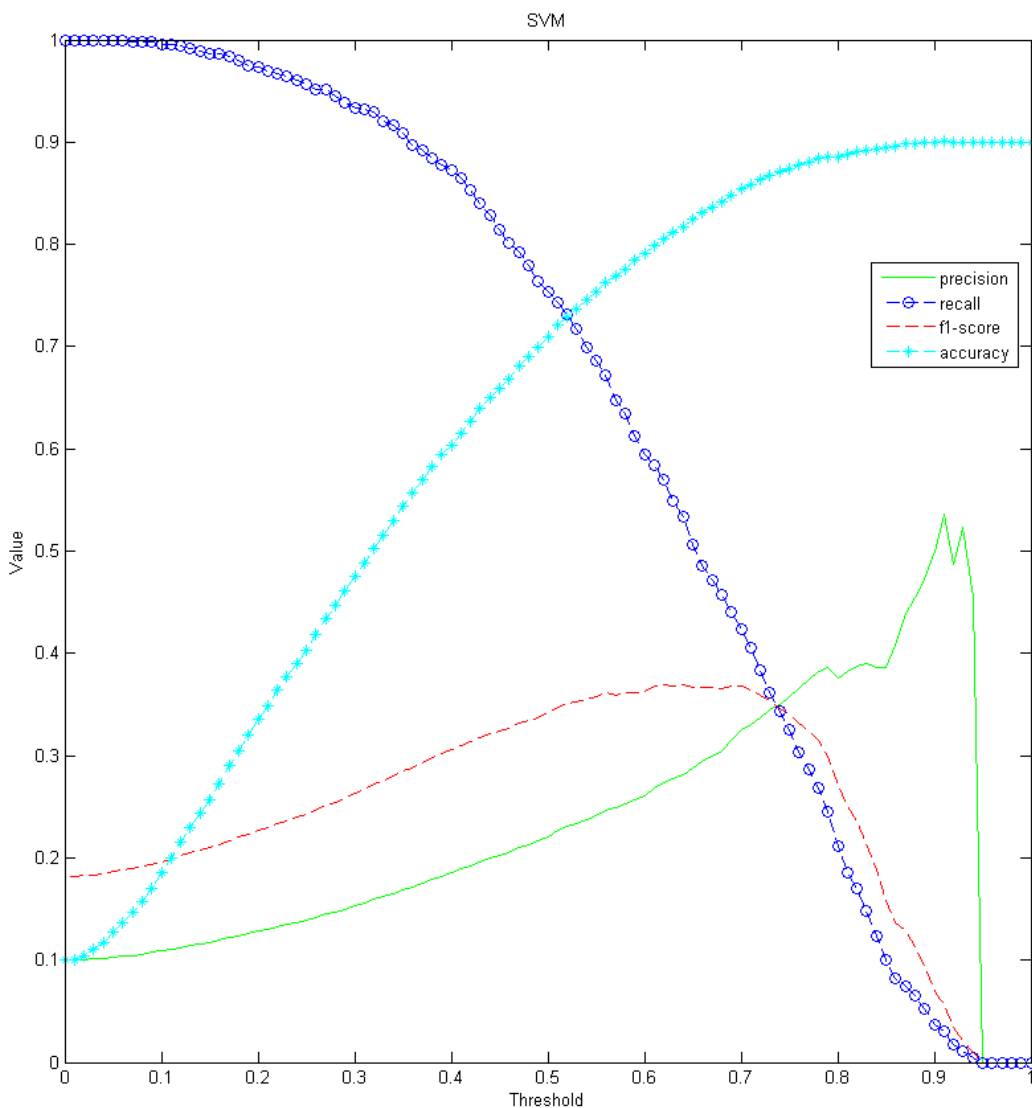
Εικόνα 4.3 Μετρικές κατά την εκπαίδευση του CNN CIFAR-10. (1 epoch = 50 iterations)

4.2 Πείραμα με σύνολο δεδομένων ελέγχου CIFAR-10

Σκοπός αυτού του πειράματος είναι η μελέτη του κατωφλίου τις πιθανότητας, στο οποίο κρίνεται εάν ένα δείγμα είναι θετικό και πως αλλάζει η απόδοση του ταξινομητή μεταβάλλοντας αυτό το κατώφλι. Ελέγχουμε όλες τις τιμές των κατωφλίων από 0 έως 1 με βήμα 0.01 για κάθε ταξινομητή ξεχωριστά και για το σύστημα ολόκληρο.

4.2.1 Μελέτη ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM

Εφαρμόζοντας το παραπάνω πείραμα στο ταξινομητή SVM παίρνουμε το γράφημα της εικόνας 5.3 .

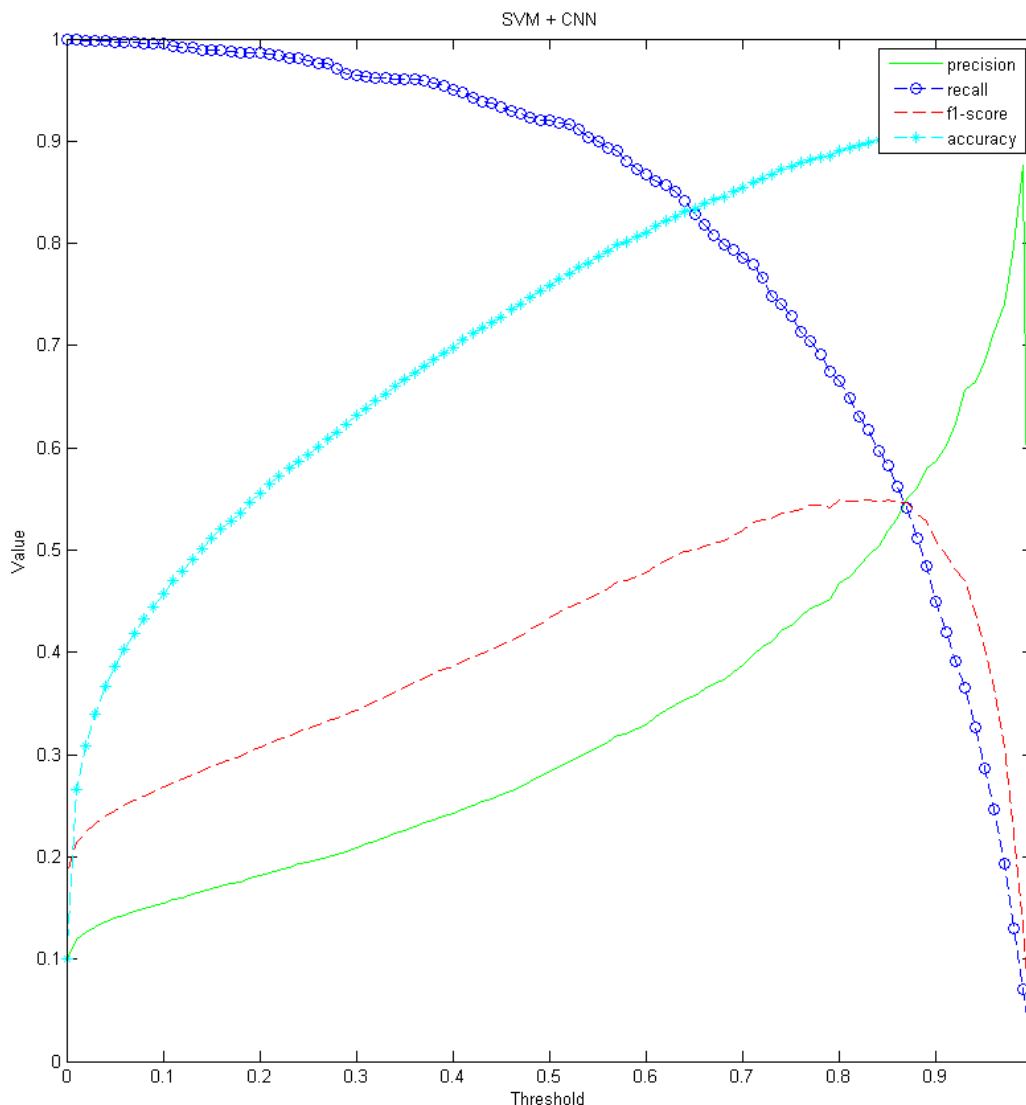


Εικόνα 4.4 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM.

Παρατηρούμε πως όσο αυξάνουμε το κατώφλι τόσο μειώνεται η ανάκληση αλλά αυξάνεται η ευστοχία. Επίσης βλέπουμε ότι τα δύο αυτά μεγέθη γίνονται ίσα όταν το κατώφλι είναι ίσο με 0.5 . Συνεπώς εάν αυξήσουμε ή μειώσουμε το κατώφλι θα έχουμε χαμηλή ανάκληση ή χαμηλή ευστοχία αντίστοιχα. Αυτό παραβιάζει τις απαιτήσεις που έχουμε από τον πρώτο ταξινομητή δηλαδή υψηλή ευστοχία και υψηλή ανάκληση. Επίσης τα λανθάνοντα θετικά δείγματα είναι αυξημένα διότι η ακρίβεια είναι χαμηλή αλλά αυτό είναι αναμενόμενο λόγω του αδύναμου ταξινομητή.

4.2.2 Μελέτη ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή CNN

Εφαρμόζοντας το παραπάνω πείραμα στο ταξινομητή CNN παίρνουμε το γράφημα της εικόνας 5.4

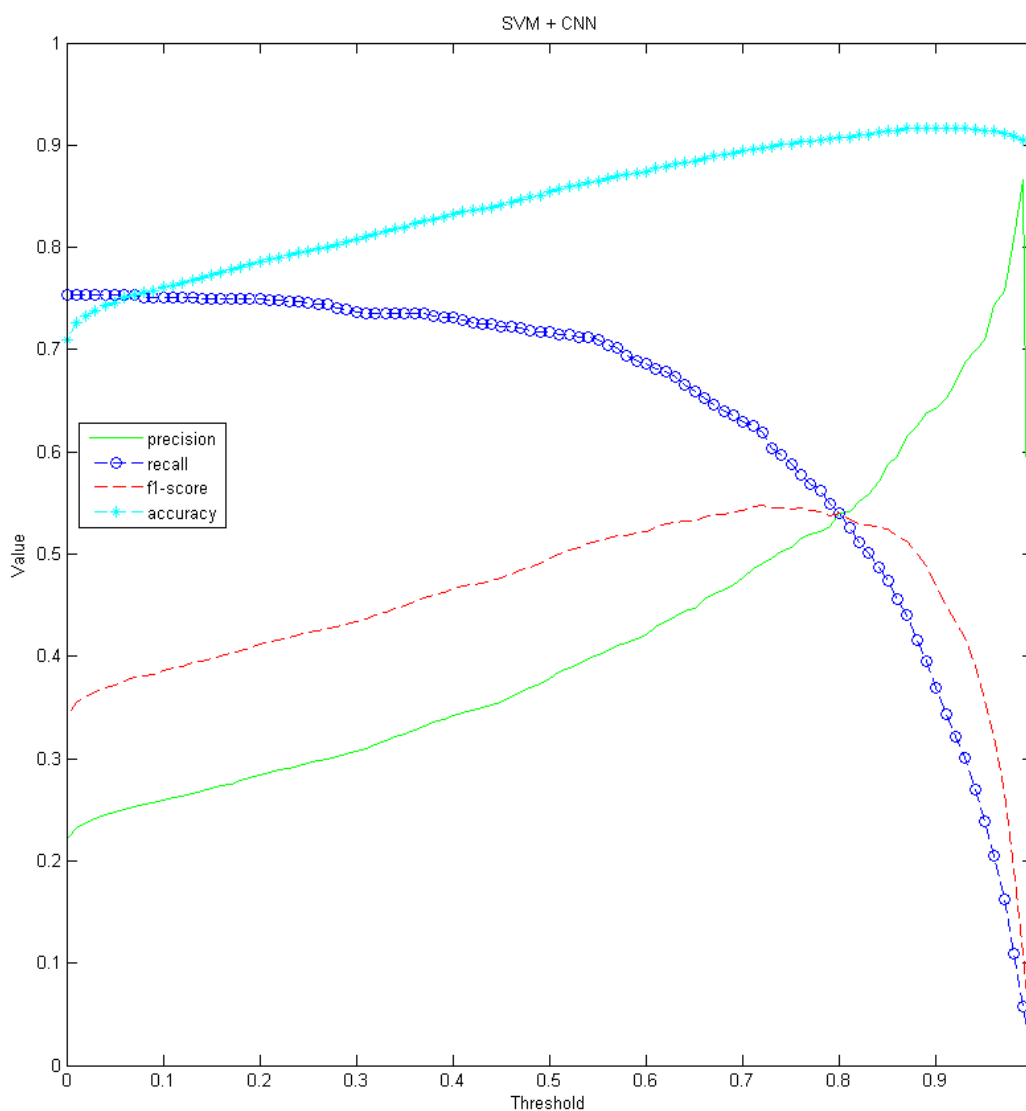


Εικόνα 4.5 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για τον ταξινομητή SVM.

Οι παρατηρήσεις ακολουθούν το ίδιο μοτίβο με το πείραμα που πραγματοποιήσαμε για το SVM. Η μόνη διαφορά είναι ότι η ευστοχία και η ανάκληση εξισώνονται για κατώφλι με τιμή κοντά στο 0.65 και σε μεγαλύτερες τιμές. Αυτό ήταν αναμενόμενο διότι ο ταξινομητής CNN είναι πιο ισχυρός από τον SVM. Αυτό όμως που δεν είναι επιθυμητό είναι ο μεγάλος αριθμός λανθανόντων θετικών δειγμάτων που φαίνεται από την αρκετά χαμηλή ακρίβεια.

4.2.3 Μελέτη ευαισθησίας κατωφλίου για το συνολικό σύστημα με χρήση CNN 2 κλάσεων

Σε αυτό το πείραμα χρησιμοποιούμε για κατώφλι του ταξινομητή SVM το 0.5 και όσα δείγματα κριθούν θετικά τα εξετάζουμε με τον ταξινομητή CNN 2 κλάσεων. Τα αποτελέσματα παρουσιάζονται στο γράφημα της εικόνας 5.5 .

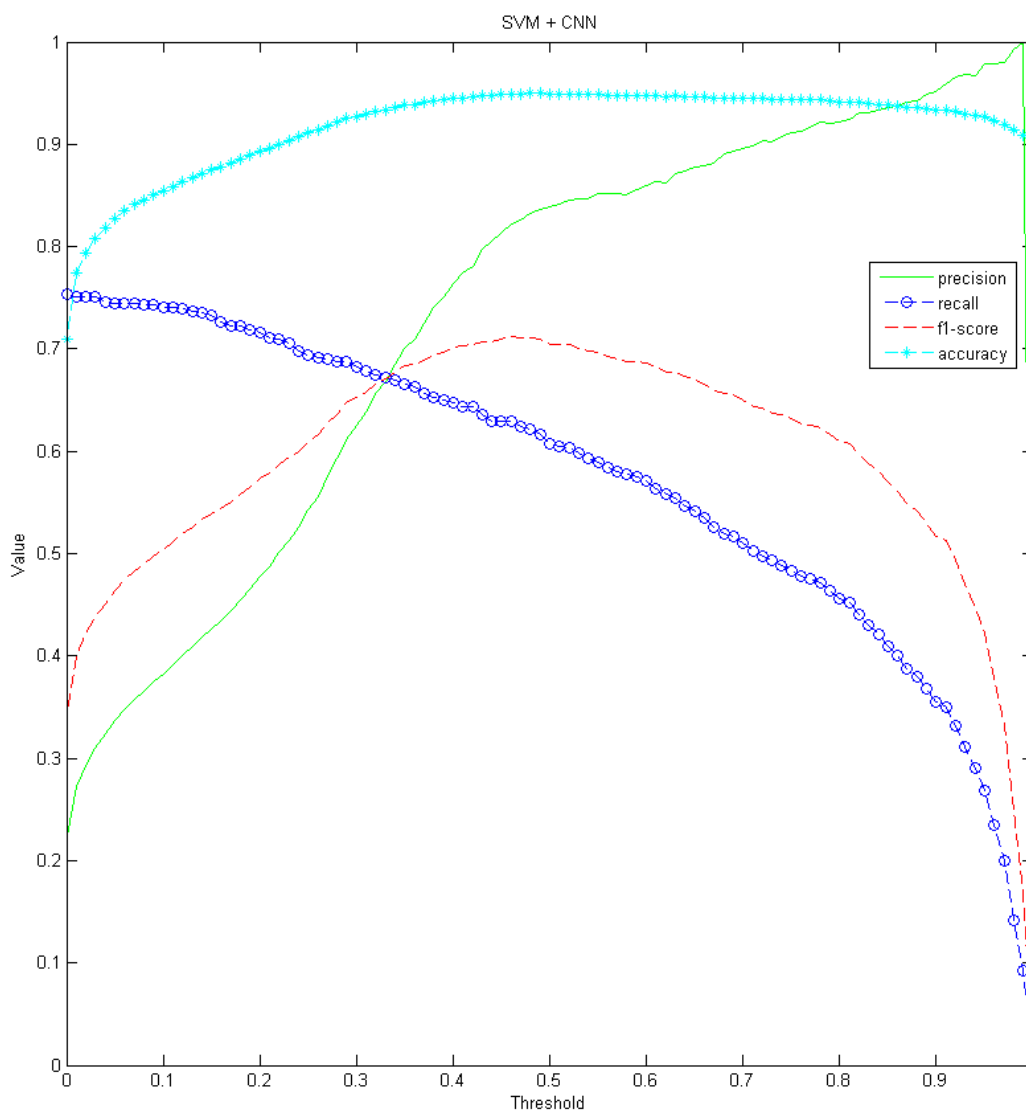


Εικόνα 4.6 Πείραμα ευαισθησίας στην τιμή του κατωφλίου για ολόκληρο το σύστημα με χρήση CNN 2 κλάσεων.

Παρατηρούμε αύξηση της ακρίβειας πράγμα αναμενόμενο και επιθυμητό αλλά δεν ξεπερνά την τιμή 0.5 ακόμα και για τιμές κατωφλίου αρκετά υψηλές, όπου η ανάκλησης παίρνει απαγορευτικά χαμηλές. Η ανάκληση μένει σε υψηλά επίπεδα, για τιμές του κατωφλίου μέχρι 0.5.

4.2.4 Μελέτη ευαισθησίας κατωφλίου για το συνολικό σύστημα με χρήση CNN 10 κλάσεων

Σε αυτό το πείραμα χρησιμοποιούμε για κατώφλι του ταξινομητή SVM το 0.5 και όσα δείγματα κριθούν θετικά τα εξετάζουμε με τον ταξινομητή CNN 10 κλάσεων. Τα αποτελέσματα παρουσιάζονται στο γράφημα της εικόνας 5.6.



Εικόνα 4.7 Πείραμα κατωφλίου για ολόκληρο το σύστημα Πείραμα κατωφλίου για ολόκληρο το σύστημα με χρήση CNN 10 κλάσεων.

Ενδιαφέρον προκαλεί το συγκεκριμένο πείραμα όπου χρησιμοποιούμε το αλγόριθμο CNN όμως αντί για 2 κλάσεις το χρησιμοποιούμε στην αρχική του μορφή, δηλαδή με 10 κλάσεις. Εάν η κλάση που θα

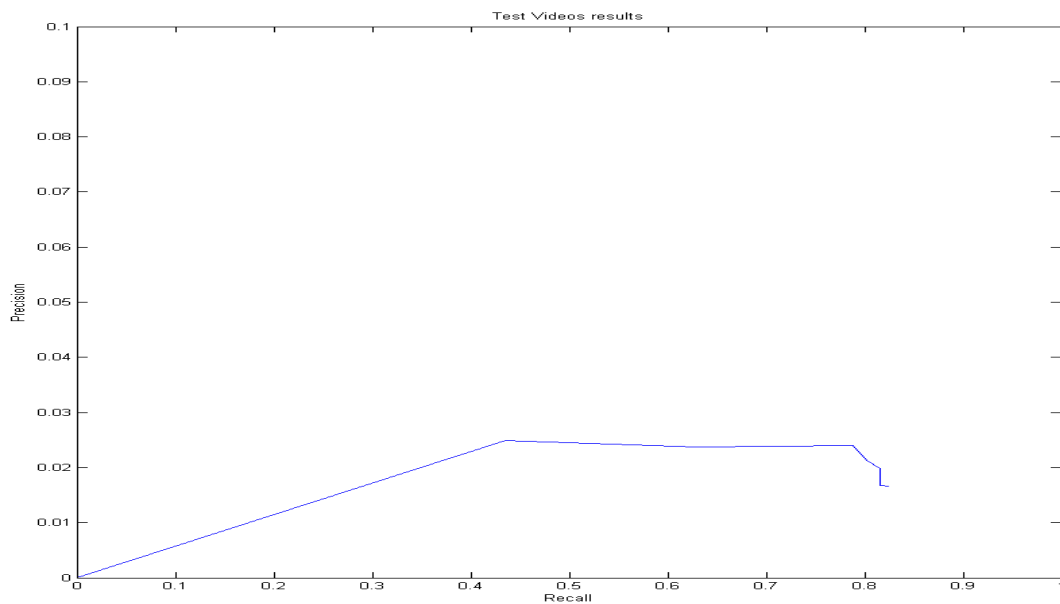
προκύψει είναι αυτή που μας ενδιαφέρει τότε το δείγμα κρίνεται θετικό, αλλιώς για οποιαδήποτε άλλη κλάση το δείγμα κρίνεται αρνητικό. Η τιμή της ακρίβειας αυξάνεται σημαντικά αλλά παρατηρούμε σημαντική πτώση στην τιμή της ανάκλησης ακόμα και για τιμές κατωφλίου πολύ χαμηλές.

4.3 Πείραμα με τυχαίες εικόνες

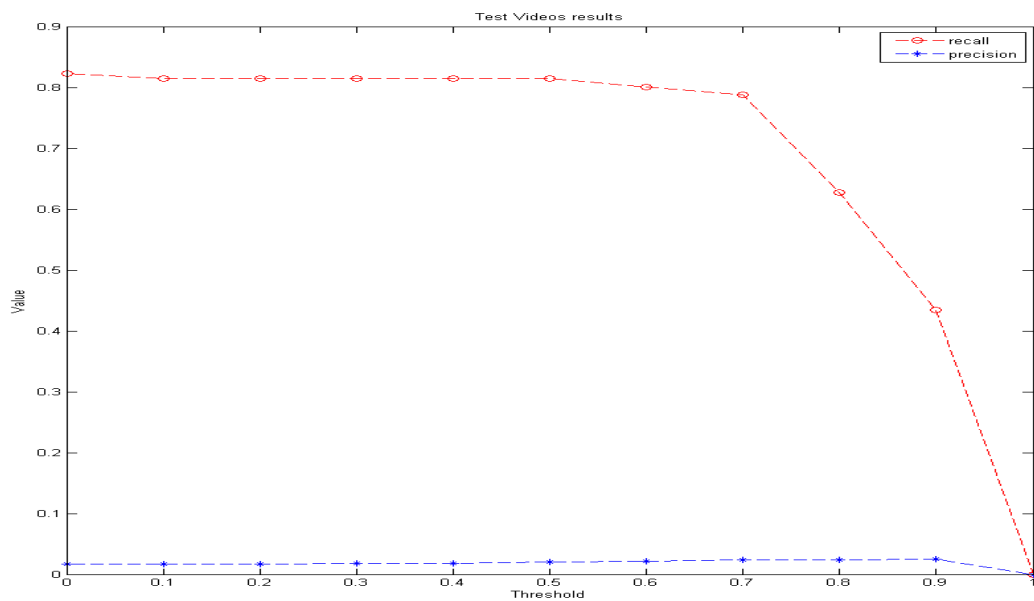
Για τις ανάγκες αυτού του πειράματος βρήκαμε 10 βίντεο τα οποία περιέχουν σκηνές με ελάφια σε διαφορετικά περιβάλλοντα και φωτισμούς. Από κάθε βίντεο συγκεντρώσαμε 100 καρτέ. Η δειγματοληψία των καρτέ γινόταν κάθε 5 καρτέ. Έτσι συγκεντρώσαμε μια ένα σύνολο δεδομένων για να εξετάσουμε τις επιδόσεις του συστήματος σε πραγματικές συνθήκες. Για κατώφλι πιθανοτήτων στο ταξινομητή SVM θέσαμε το 0.5, και μελετήσαμε τα αποτελέσματα του συστήματος μεταβάλλοντας το κατώφλι του ταξινομητή CNN 2 κλάσεων. Ο λόγος που δεν επιλέξαμε τον ταξινομητή CNN 10 κλάσεων είναι η χαμηλή ανάκληση πράγμα που φάνηκε από την αρχή του πειράματος αυτού, καθώς οι αληθώς θετικές ανιχνεύσεις ήταν σχεδόν μηδενικές.

Οι τιμές των κατωφλίων που μελετήθηκαν είναι από 0 έως 1 με βήμα 0.1. Ο λόγος για το τόσο υψηλό βήμα οφείλεται στο ότι η επαλήθευση των αληθώς θετικών ανιχνεύσεων έπρεπε να γίνει με φυσικά μέσα πράγμα το οποίο απαιτεί πολύ χρόνο. Τα αποτελέσματα των πειραμάτων παρουσιάζονται στα διαγράμματα των εικόνων 5.7 και 5.8.

Αυτό που παρατηρούμε από τα γραφήματα είναι ότι η ανάκληση έχει υψηλές τιμές συνεπώς έχουμε πολλά αληθώς θετικά δείγματα. Παρόλα αυτά η ακρίβεια κινείται σε πολύ χαμηλές τιμές πράγματα που σημαίνει ότι υπάρχουν πάρα πολλά λανθάνοντα θετικά δείγματα. Αυτό επιβεβαιώνεται βλέποντας και τις εικόνες, 5.9 και 5.10. Ιδιαίτερα σε σημεία με βλάστηση οι ταξινομητές δείχνουν να μην μπορούν να κατηγοριοποιήσουν σωστά τα δείγματα. Αξίζει να σημειωθεί πως όσο και να αυξάνουμε το κατώφλι του ταξινομητή CNN δεν παρατηρούμε αύξηση της ακρίβειας πράγμα που σημαίνει ότι τα λανθάνοντα θετικά δείγματα κατηγοριοποιούνται ως θετικά από τον ταξινομητή με πολύ μεγάλη πιθανότητα, και συχνά μεγαλύτερη των αληθώς θετικών δειγμάτων. Αυτό οφείλεται στην μη καταλληλότητα του συνόλου δεδομένων που χρησιμοποιήσαμε. Η ύπαρξη εικόνων ενός ζώου σε πολλές διαφορετικές στάσεις σώματος και η λήψη των εικόνων οι οποίες εστιάζουν άλλοτε στο κεφάλι και άλλοτε σε ολόκληρο το σώμα “μπερδεύουν” τους ταξινομητές και παράγουν αυτά τα αποτελέσματα. Επίσης η απουσία αρνητικών δειγμάτων από το σύνολο δεδομένων που να περιλαμβάνουν βλάστηση από κοντινή λήψη είναι ένας ακόμη παράγοντας αυτής της συμπεριφοράς.



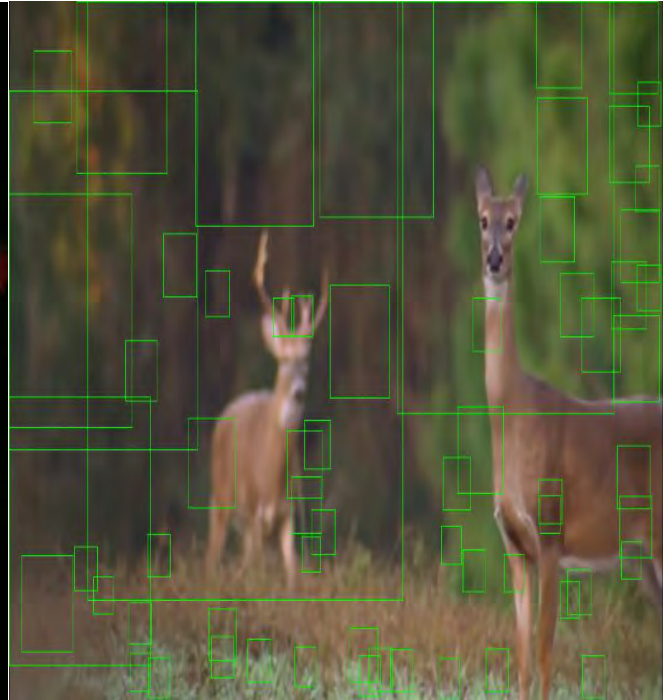
Εικόνα 4.8 Precision-Recall στο πείραμα τυχαίων εικόνων.



Εικόνα 4.9 Τιμές Precision και Recall ως προς την τιμή του κατωφλίου.



Εικόνα 4.11 Ανιχνεύσεις συστήματος σε νυχτερινή φωτογραφία



Εικόνα 4.11 Ανίχνευση συστήματος σε περιοχές με βλάστηση

4.4 Χρόνοι εκτέλεσης

4.4.1 Ανάλυση χρόνων για κλιμακωτή ανίχνευση

Τα μηχανήματα στα οποία έγιναν οι μετρήσεις διέθεταν τα παρακάτω κατακτηριστικά:

CPU:

- Architecture: x86_64
- CPU op-mode(s): 32-bit, 64-bit
- Byte Order: Little Endian
- CPU(s): 8
- On-line CPU(s) list: 0-7
- Thread(s) per core: 2
- Core(s) per socket: 4
- Socket(s): 1
- NUMA node(s): 1
- Vendor ID: GenuineIntel
- CPU family: 6
- Model: 62
- Stepping: 4

- CPU MHz: 3701.000
- BogoMIPS: 7404.18
- Virtualization: VT-x
- L1d cache: 32K
- L1i cache: 32K
- L2 cache: 256K
- L3 cache: 10240K
- NUMA node0 CPU(s): 0-7

GPU:

- GTX 690 GPU Engine Specs:
- 3072 CUDA Cores
- 915 Base Clock (MHz)
- 1019 Boost Clock (MHz)
- 234-Texture Fill Rate (billion/sec)
- GTX 690 Memory Specs:
- 6.0 Memory Speed (Gbps)
- 4096 MB (2048 MB per GPU) GDDR5Standard Memory Config
- 512-bit (256-bit per GPU)Memory Interface Width
- 384 Memory Bandwidth (GB/sec)

Η αρχιτεκτονική μας βασίζεται στην υπόθεση ότι ο ταξινομητής SVM εκτελείται σε λιγότερο χρόνο από τον ταξινομητή CNN. Σκοπός είναι ο χρόνος που απαιτείται για την κλιμακωτή ανίχνευση μιας εικόνας, από το σύστημα δύο σταδίων, να μην είναι μεγαλύτερος από τον χρόνο που θα απαιτούνταν εάν το σύστημα διέθετε μόνο τον ταξινομητή CNN.

Οι μετρήσεις οι οποίες πήραμε από τα πειράματα μας δείχνουν πως ο ταξινομητής SVM, μαζί με τον περιγραφέα HOG, είναι **800** φορές ταχύτερος από τον ταξινομητή CNN(εικόνα 4.12). Για να αναλύσουμε περισσότερο αυτά τα μεγέθη θα χρησιμοποιήσουμε ένα παράδειγμα.

Για μια εικόνα με διαστάσεις HD 1280x720 εικονοστοιχεία η διαδικασία της κλιμακωτής αναζήτησης, με βήμα 8 εικονοστοιχεία στον οριζόντιο και κατακόρυφο άξονα και ρυθμό κλιμάκωσης 1.05, θα εκτελέσει συνολικά 138,243 ανιχνεύσεις. Από αυτές όσες θεωρηθούν θετικές, και μόνο αυτές, θα δοθούν σαν είσοδο στον ταξινομητή CNN. Για κάθε τέτοια ανίχνευση ο ταξινομητής SVM χρειάζεται 0.2 msec ο ταξινομητής CNN χρειάζεται 160 msec. Εάν το πρώτο στάδιο της ανίχνευσης γίνει με τον ταξινομητή SVM θα χρειαστούν 27.648 msec ενώ εάν χρησιμοποιηθεί ο CNN θα χρειαστούν 22.119 sec.

4.4.2 Ομαδοποίηση θετικών ανιχνεύσεων του 1^{ου} σταδίου

Επιπροσθέτως, κατά την διαδικασία την κλιμακωτής ανίχνευσης, πολλές θετικές ανιχνεύσεις που προκύπτουν απευθύνονται στο ίδιο αντικείμενο. Η αιτία αυτού του φαινομένου είναι ότι μια ανίχνευση μπορεί να κριθεί και ως θετική ακόμα και εάν το αντικείμενο δεν βρίσκεται στο κέντρο του παραθύρου που σαρώνει την εικόνα. Ταυτόχρονα θετικές ανιχνεύσεις που δεν διαθέτουν πολλές θετικές ανιχνεύσεις σε κοντινή απόσταση από αυτές μπορούν να κριθούν ως λανθάνουσες θετικές ανιχνεύσεις.

Την απομάκρυνση αυτών των περιπτώσεων ή λανθάνουσών θετικών ανιχνεύσεων αναλαμβάνει ο αλγόριθμος μέσης μετατόπισης (Mean Shift [27]) κάνοντας ομαδοποίηση (*grouping*). Η ύπαρξη αυτού του αλγορίθμου είναι ανεξάρτητη από τον ταξινομητή που θα χρησιμοποιηθεί στο πρώτο στάδιο και αφορά αποκλειστικά την κλιμακωτή ανίχνευση.

Ο αλγόριθμος μέσης μετατόπισης δεν είναι υλοποιημένος σε GPU, και έχει σαν μέσο χρόνο εκτέλεσης 3.91 sec. Αυτό δημιουργεί συμφόρηση στο συγκεκριμένο σημείο και καθυστερεί όλη την διαδικασία της ανίχνευσης.

Κεφάλαιο 5 Συμπεράσματα και μελλοντικοί στόχοι

Στην παρούσα εργασία μελετήθηκε και υλοποιήθηκε ένα σύστημα ανίχνευσης ζώων σε φυσικά τοπία. Στο σύστημα αυτό η ανίχνευση γινόταν σε δύο στάδια από δύο διαφορετικούς ταξινομητές. Μέσα από την υλοποίηση αυτού του συστήματος και την επίλυση των προβλημάτων που προέκυψαν, μελετήθηκαν και κατανοήθηκαν, σε μεγάλο βαθμό, οι παράμετροι που διέπουν τους αλγόριθμους υπολογιστικής όρασης.

Η ενσωμάτωση του περιγραφέα HOG [1] σε έναν ταξινομητή μας βοήθησε να αντιληφθούμε την σημασία και τον ρόλο που έχουν οι διαφορετικές απεικονίσεις μια εικόνας σε προβλήματα κατηγοριοποίησης. Η κατανόηση και η χρήση του αλγορίθμου SVM [2] μας έδειξε τη λειτουργία ενός γραμμικού ταξινομητή και ανέδειξε τις δυνατότητες και τις αδυναμίες του. Αντίστοιχα η ενασχόλησή μας με συνελκτικά νευρωνικά δίκτυα CNN [3] μας έδωσαν μια καλύτερη κατανόηση της λειτουργίας των νευρωνικών δικτύων. Η κατασκευή ενός συστήματος ανίχνευσης ζώων αποδείχθηκε ένα πολύ δύσκολο έργο με πολλές καινούργιες παραμέτρους οι οποίες πρέπει να ληφθούν υπόψιν εάν θέλουμε να βελτιώσουμε τα αποτελέσματα μας

Το σύστημα που αναπτύξαμε μας έδωσε υψηλά ποσοστά στην αναγνώριση αληθώς θετικών δειγμάτων. Δυστυχώς όμως το πλήθος των λανθανόντων θετικών δειγμάτων ήταν υπερβολικά υψηλό. Η βασική αιτία αυτού του προβλήματος ήταν το μη κατάλληλο σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση των ταξινομητών. Ένα σωστό, για την χρήση που το θέλουμε, σύνολο δεδομένων θα έπρεπε να περιέχει εικόνες με ζώα σε μία συγκεκριμένη στάση σώματος και η λήψη να έχει γίνει από μια συγκεκριμένη οπτική γωνία. Ταυτόχρονα θα έπρεπε να περιέχει αρνητικά δείγματα τα οποία να απεικονίζουν το φυσικό περιβάλλον των ζώων που επιθυμούμε να ανιχνεύσουμε.

Παρόλα τα αρνητικά αποτελέσματα στο κομμάτι της ανίχνευσης, η αρχική μας υπόθεση για την τρόπο που έπρεπε να δομηθεί ένα τέτοιο σύστημα, δηλαδή την επιλογή του ταξινομητή SVM για το πρώτο στάδιο της κατηγοριοποίησης αποδείχθηκε σωστή. Η υλοποίηση των αλγορίθμων HOG και SVM είναι κατά **800** φορές πιο γρήγορη από την υλοποίηση του ταξινομητή CNN βελτιώνοντας έτσι τον χρόνο που απαιτεί το σύστημα για την κατηγοριοποίηση δειγμάτων. Η ταξινόμηση μιας εικόνας HD από τον ταξινομητή CNN χρειάζεται περίπου 22 δευτερόλεπτα, πράγμα που τον καθιστά απαγορευτικό για εφαρμογές πραγματικού χρόνου.

Μια πρόταση για την συνέχεια της παρούσας εργασίας θα ήταν η κατασκευή ενός συνόλου δεδομένων από την αρχή με τις προδιαγραφές που θεωρούμε ότι απαιτούνται για την σωστή λειτουργία ενός τέτοιου συστήματος. Επίσης θα μπορούσαν να δοκιμαστούν και διαφορετικοί ταξινομητές για το πρώτο στάδιο της κατηγοριοποίησης καθώς και ένα καινούργιο συνελκτικό νευρωνικό δίκτυο με διαφορετική δομή. Απαραίτητη όμως κρίνουμε την βελτιστοποίηση του αλγορίθμου μέσης μετατόπισης ή την αντικατάσταση του εάν θέλουμε να πετύχουμε εκτέλεση του συστήματος σε πραγματικό χρόνο.

Βιβλιογραφία/Αναφορές

- [1] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.
- [2] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992.
- [3] Cirosan, Dan C., et al. "Flexible, high performance convolutional neural networks for image classification." *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*. Vol. 22. No. 1. 2011.
- [4] Tardif, Louis-Paul. "Collisions involving motor vehicles and large animals in Canada." Unpublished report prepared for the Transport Canada Road Safety Directorate (2003).
- [5] Williams, Allan F., and Joann K. Wells. "Characteristics of vehicle-animal crashes in which vehicle occupants are killed." *Traffic Injury Prevention* 6.1 (2005): 56-59.
- [6] Israel, Martin. "A UAV-based roe deer fawn detection system." *International Archives of Photogrammetry and Remote Sensing* 38 (2011): 1-5.
- [7] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).
- [8] Bradski, Gary. "The OpenCV Library (2000)." *Dr. Dobb's Journal of Software Tools* (2000).
- [9] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014.

[10] Hosmer Jr, David W., and Stanley Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[11] Haykin, Simon. "Neural Network: A comprehensive foundation." *Neural Networks* 2.2004 (2004).

[12] Kramer, Oliver. "K-Nearest Neighbors." *Dimensionality Reduction with Unsupervised Nearest Neighbors*. Springer Berlin Heidelberg, 2013. 13-23.

[13] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." *Journal of the Royal Statistical Society. Series C (Applied Statistics)* 28.1 (1979): 100-108.

[14] Sharafsaleh, Mohammad, et al. *Evaluation of an Animal Warning System Effectiveness*. No. UCB-ITS-PRR-2010-22. 2010.

[15] Holland, Tiffany D., et al. *Animal-vehicle Crash Mitigation Using Advanced Technology: Phase II: System Effectiveness and System Acceptance*. No. FHWA-RD-TPF-09-14. Oregon Department of Transportation, Research Unit, 2009.

[16] Zhou, Debao, and Jingzhou Wang. "Deer Detection in Thermal Images for Traffic Safety Using Contour Based Histogram of Oriented Gradient Method." *Modern Traffic and Transportation Engineering Research* 2.1 (2013): 34-40.

[17] Mammeri, Abdelhamid, et al. "An efficient animal detection system for smart cars using cascaded classifiers." *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014.

[18] Pietikainen, M., A. Hadid, and T. Ahonen. "Computer Vision Using Local Binary Patterns vol. 40: Springer." *New York City, USA* (2011).

[19] Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Computational learning theory*. Springer Berlin Heidelberg, 1995.

[20] Matuska, Slavomir, et al. "A novel system for automatic detection and classification of animal." *ELEKTRO, 2014*. IEEE, 2014.

[21] Lowe, David G. "Object recognition from local scale-invariant features." *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. Vol. 2. Ieee, 1999.

[22] Bay, Herbert, et al. "Speeded-up robust features (SURF)." *Computer vision and image understanding* 110.3 (2008): 346-359.

[23] Israel, Martin. "A UAV-based roe deer fawn detection system." *International Archives of Photogrammetry and Remote Sensing* 38 (2011): 1-5.

[24] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: a library for support vector machines." *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011): 27.

[25] Deng, Li, and Dong Yu. "Deep learning: Methods and applications." *Foundations and Trends in Signal Processing* 7.3-4 (2014): 197-387.

[26] "A factsheet on Volvo Cars' safety technology in the new Volvo S90" 2015. [Online]. Available: <https://www.media.volvocars.com/global/en-gb/download/170199>

[27] Cheng, Yizong. "Mean shift, mode seeking, and clustering." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 17.8 (1995): 790-799.

[28] Chris McCormick "HOG Person Detector Tutorial" [Online]. Available: <https://chrisjmcormick.wordpress.com/2013/05/09/hog-person-detector-tutorial/>

[29] Introduction to Support Vector Machines [*Online*]. Available: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html

[30] Deep Learning for Vision with Caffe [*Online*]. Available: https://docs.google.com/presentation/d/1UeKXVgRvVxg9OUdh_UiC5G71UMscNPlvArsWER41PsU/edit#slide=id.p

[31] Neural Networks [*Online*]. Available: http://docs.opencv.org/2.4/modules/ml/doc/neural_networks.html

[32] Urban and Natural Scene Categories [*Online*]. Available: <http://cvcl.mit.edu/database.htm>