



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

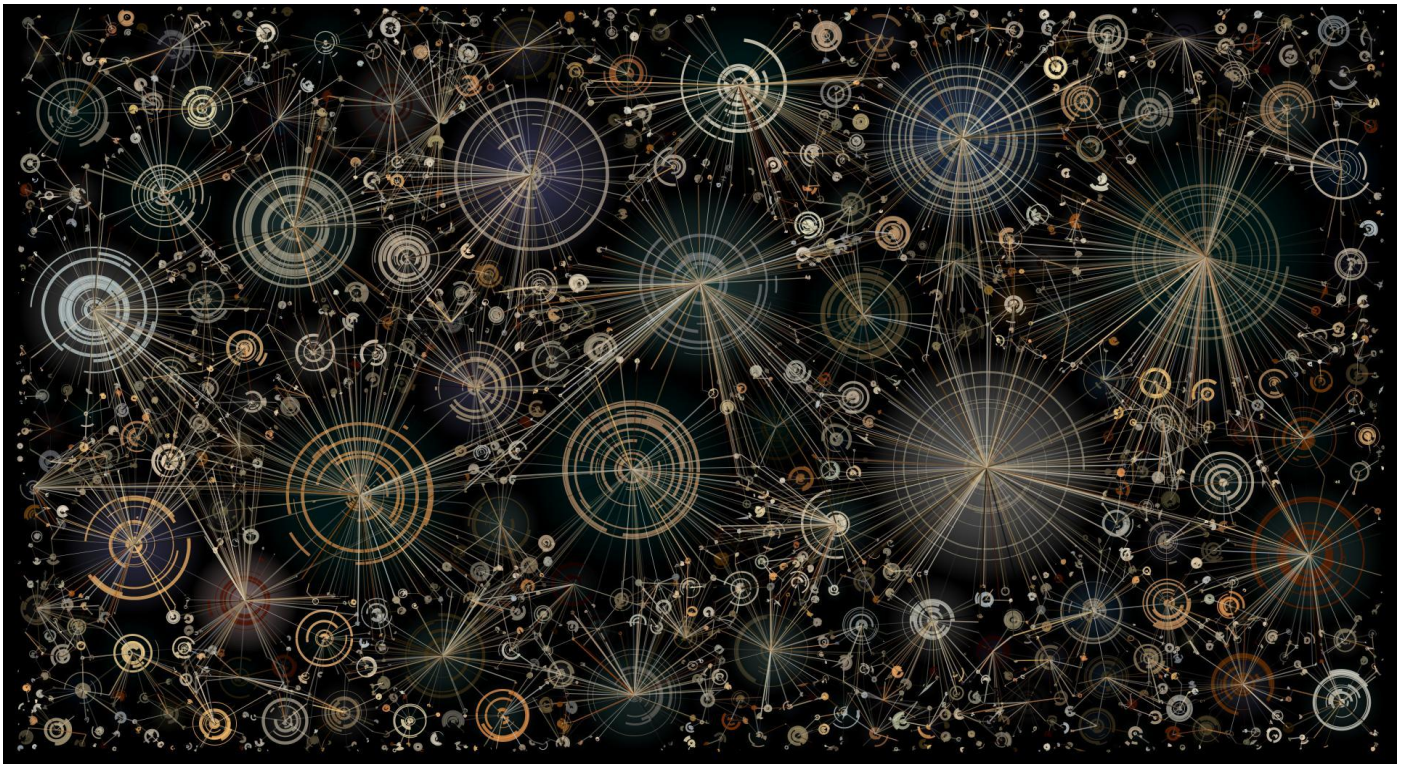
Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής

Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας

Εργαστηριακό Πρότζεκτ – Μέρος Β

Αριστείδης Ανδρουτσόπουλος

AM: 1054274



52940_packing_art_031 (artistic) 14.08.24/05/2013

Ακαδημαϊκό Έτος 2022-2023

Περιεχόμενα

1. Εισαγωγή.....	3
1.1. Περιγραφή Άσκησης.....	3
2. Ασκήσεις.....	3
2.1. CNN - Συνελικτικά Νευρωνικά Δίκτυα.....	3
2.1.1. Ερώτημα 1: Δεδομένα Εκμάθησης	3
2.1.2. Ερώτημα 2: SGD για CNN.....	4
2.1.3. Ερώτημα 3: Σφάλμα Κατηγοριοποίησης και Ακρίβεια	6
2.1.4. Ερώτημα 4: Confusion Matrix	7
2.2. HOG – Support Vector Machines	8
2.2.1. Ερώτημα 1: In-Built HOG and SVM.....	8
2.2.2. Ερώτημα 2: Αποτελέσματα – HOG Patches	9
2.2.3. Ερώτημα 3: Confusion Matrix	11
2.2.4. Ερώτημα 4: Bonus: Custom HOG	12
3. Κώδικες Υλοποίησης.....	13
References	14

1. Εισαγωγή

1.1. Περιγραφή Άσκησης

Στο παρόν Μέρος Β – **Θέμα 1** του εργαστηριακού πρότζεκτ ζητείται η εφαρμογή κάποιων αλγορίθμων εκμάθησης για την κατηγοριοποίηση εικόνων. Συγκεκριμένα ζητείται η χρήση Συνελικτικών Νευρωνικών Δικτύων καθώς και χρήση περιγραφέα χαρακτηριστικών HOG (Histogram of Oriented Gradients) σε συνδιασμό με SVM (Support Vector Machines). Για την υλοποίηση των αλγορίθμων επιλέχθηκε ο χώρος εργασίας και το εργαλείο MATLAB με τους αντίστοιχους κώδικες για κάθε ερώτημα να βρίσκονται ανεβασμένα στο εργαλείο [GitHub - Σελίδα GitHub](#).

2. Ασκήσεις

2.1. CNN - Συνελικτικά Νευρωνικά Δίκτυα

Προβλήματα επιβλεπόμενης μάθησης, όπως είναι η εκπαίδευση ενός CNN νευρωνικού απαιτεί την προσθήκη ετικέτας κατηγορίας στην διαστατικότητα του κάθε στοιχείου. Συγκεκριμένα για το δοσμένο dataset, κάθε στοιχείο περιέχει μια εικόνα σε μορφή μητρώου δύο διαστάσεων καθώς και μία ετικέτα για την κατηγορία στην οποία ανήκει η εικόνα αυτή. Η διαδικασία εκμάθησης, όπως αναφέρεται στην εκφώνηση [1], ακολουθεί στοχαστική βελτιστοποίηση προς Minima του Expectation μιας συνάρτησης κόστους που ορίζει το κόστος κάθε λανθασμένης κατηγοριοποίησης. Για την προσέγγιση του E(.) χρησιμοποιείται ο αριθμητικός μέσος των διαθέσιμων δεδομένων, τα οποία έχουν παραχθεί από την κατανομή $P(x,y)$ που περιγράφει στατιστικά τα δεδομένα του σετ εικόνων.

2.1.1. Ερώτημα 1: Δεδομένα Εκμάθησης

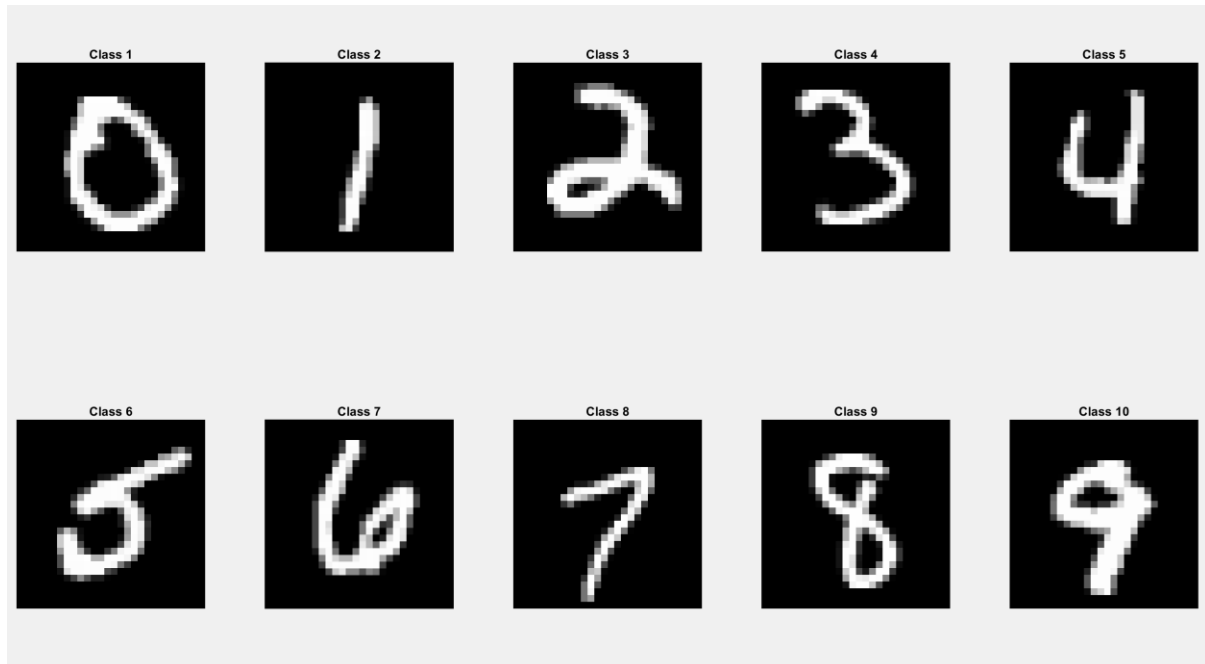
Τα δεδομένα εκμάθησης και ελέγχου προέρχονται από την συλλογή Minst [2] και περιέχει εικόνες ψηφίων, γραμμένα στο χέρι, σε διάφορες παραλλαγές γραφικού χαρακτήρα. Οι εικόνες είναι ασπρόμαυρες μεγέθους 28x28 και χωρίζονται σε δύο υποσύνολα, για εκπαίδευση και για έλεγχο, μεγέθους 60.000 και 10.000 αντίστοιχα. Κάθε στοιχείο-εικόνα συνοδεύεται από αντίστοιχη ετικέτα του αριθμού που αντιπροσωπεύει το γραπτό ψηφίο. Για την εύκολη και γρήγορη διαχείριση των δεδομένων έγινε χρήση της μετατροπής [3] ενώ ένας άλλος τρόπος παρόμοιος διαχείρισης μπορεί να βρεθεί στο script.[4]

Εισάγοντας τα δεδομένα μέσω της [3] δημιουργούνται στην Matlab τα εξής αντικείμενα:

training		test		training	
1x1 struct with 5 fields				1x1 struct with 5 fields	
Field	Value	Field	Value	Field	Value
count	10000	count	60000	count	60000
width	28	width	28	width	28
height	28	height	28	height	28
images	28x28x10000 double	images	28x28x60000 double	images	28x28x60000 double
labels	10000x1 double	labels	60000x1 double	labels	60000x1 double

Εικόνα 1: Τα αντικείμενα των δεδομένων εκμάθησης και δοκιμής

Για την σωστή χρήση του συνόλου εικόνων προστίθεται ακόμα μία διάσταση καθώς η συνάρτηση εκπαίδευσης της Matlab απαιτεί 4D είσοδο (Αριθμός Επιπέδων). Παρακάτω εμφανίζονται μία εικόνα για κάθε κατηγορία - τυχαία επιλογή αντίστοιχης εικόνας σε κάθε code run-:



Εικόνα 2: Τυχαία δείγματα εικόνων από την κάθε κλάση.

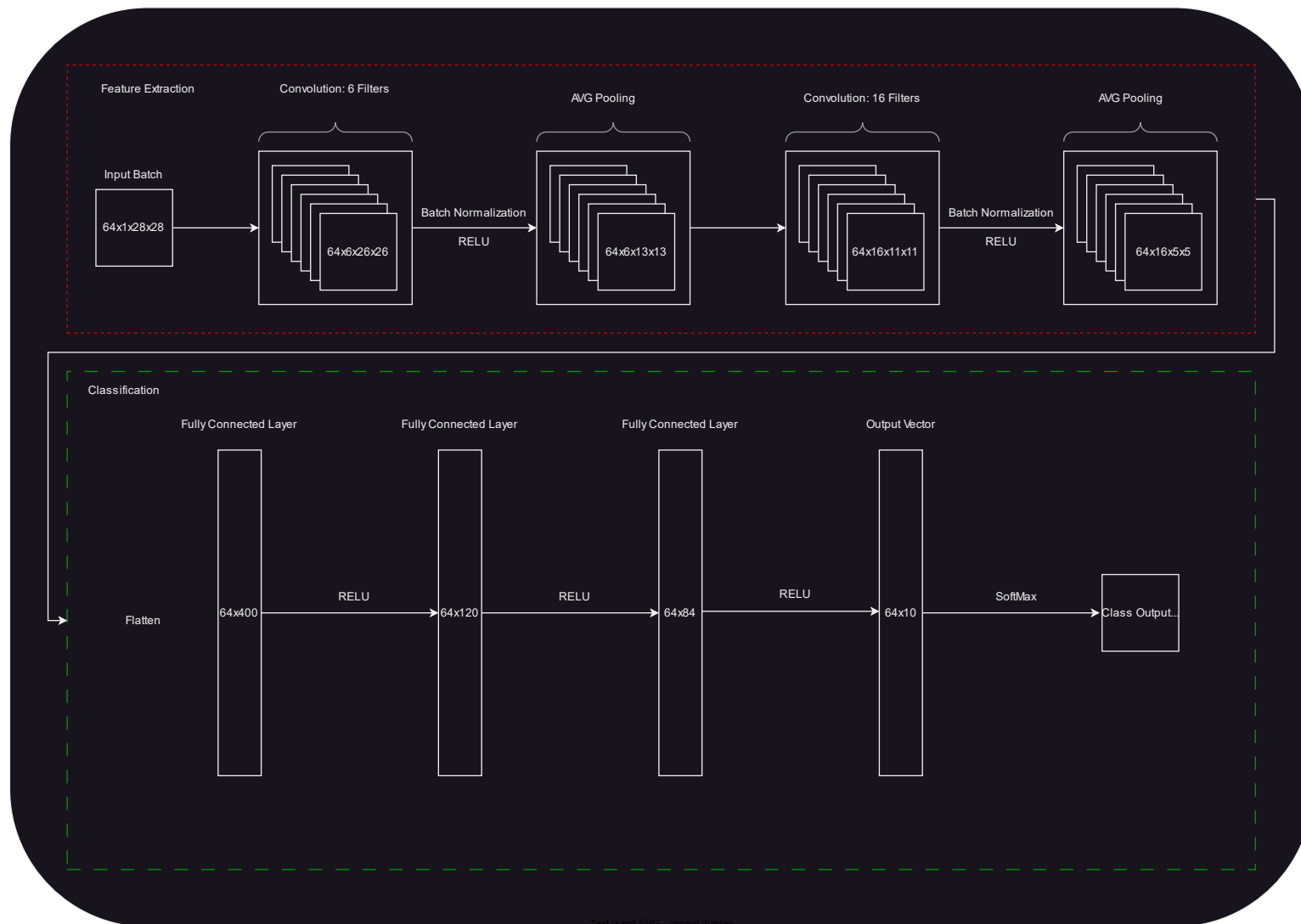
2.1.2. Ερώτημα 2: SGD για CNN

Για την υλοποίηση του SGD και του CNN έγινε χρήση των βιβλιοθηκών της Matlab. Η διαδικασία ελαχιστοποίησης του Expectation με την βοήθεια SGD έγινε με τις εξής παραμέτρους[5]:

- Momentum: 0.9 (Default) – Αδράνεια βήματος όταν ο Gradient κινείται στην ο Gradient έχει ίδια κατεύθυνση για μεγάλα χρονικά διαστήματα.
- MaxEpochs: 3
- MiniBatch size: 64
- Learning Rate: 0.01 – Μέγεθος βήματος προς σύγκλιση.

Ενώ για τις γραφικές παραστάσεις εξέλιξης της εκπαίδευσης τέθηκε Validation Frequency κάθε 30 επαναλήψεις. Η κυρίαρχη συνάρτηση ενεργοποίησης εσωτερικά του νευρωνικού είναι η RELU ενώ η τελική συνάρτηση ενεργοποίησης είναι η SoftMax (Επιλογή 1 τιμής εξόδου).

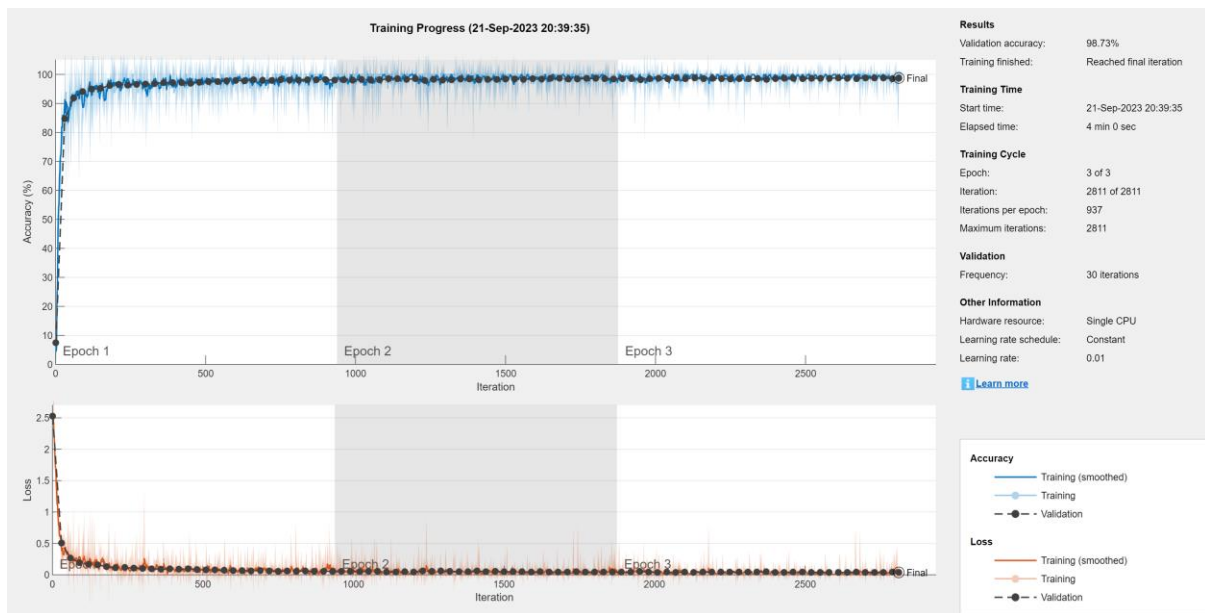
Το νευρωνικό δίκτυο CNN βασίζεται στην διαδικασία δημιουργίας activation maps μέσω συνελίξεων ενός αρχικού τένσορα με φίλτρα που στην περίπτωση μας είναι μεγέθους 3x3, χωρίς Padding και με αναπήδηση ανά μία στήλη (stride). Στην συνέχεια γίνεται χρήση Στατιστικού Μέσου Pooling μεγέθους 2x2 και αναπήδηση ανα δύο στήλες για την σμίκρυνση του κάθε Map. Αυτή η διαδικασία επαναλαμβάνεται δύο φορές με διαφορετικό αριθμό φίλτρων μέχρις ότου τα δεδομένα που απομένουν είναι σχετικά μικρά σε μέγεθος για να διανυσματοποιηθούν (Vectorization) ώστε να δημιουργηθεί μονοδιάστατη έξοδος κατηγοριοποίησης. Καθόλη την διαδικασία χρησιμοποιείται batch normalization για την κανονικοποίηση των διανυσμάτων ενεργοποίησης με χρήση των ροπών 1^{ης} και 2^{ης} τάξης (mean, variance)[6]. Για την υλοποίηση του χρησιμοποιήθηκαν διάφορες πηγές όπως [7] [8]. Για την καλύτερη κατανόηση της διαδικασίας παρουσιάζεται μια γραφική απεικόνιση που δείχνει την γενική μορφή της, βασισμένη στην απεικόνιση που υπάρχει στην δημοσίευση [9].



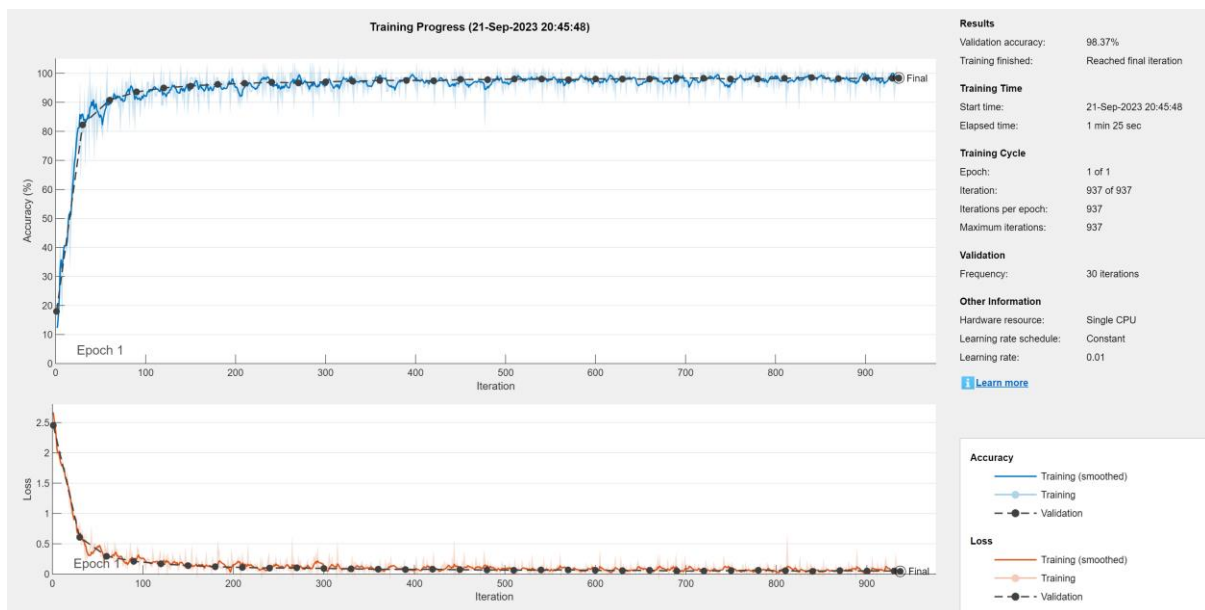
Εικόνα 3: Γραφική αναπαράσταση του Νευρωνικού Δικτύου, μαζί με τα μεγέθη τένσορα.

2.1.3. Ερώτημα 3: Σφάλμα Κατηγοριοποίησης και Ακρίβεια

Τα αποτελέσματα της κατηγοριοποίησης είναι πολύ ικανοποιητικά. Το μοντέλο μας συγκλίνει σε ικανοποιητικό βαθμό σε μικρό αριθμό εποχών ενώ δεν παρατηρείται κάποια Υπερεκπαίδευση. Παρακάτω εμφανίζονται τα αποτελέσματα για 3 εποχές και 1 εποχή, με τις γραφικές παραστάσεις για Loss και Precision αντίστοιχα. Είναι εμφανές ότι θα ήταν χρήσιμη η προσθήκη Validation Stopping (Early Stopping) καθώς το μοντέλο μας φαίνεται να συγκλίνει σε πολύ μικρό αριθμό επαναλήψεων.



Εικόνα 4: 3 Εποχές - Γραφικές Παραστάσεις για Loss και Accuracy



Εικόνα 5: 1 Εποχή - Γραφικές Παραστάσεις για Loss και Accuracy

Γίνεται άμεσα αισθητό πως το συγκεκριμένο μοντέλο προσεγγίζει το πρόβλημα σε έναν πολύ καλό βαθμό. Με μόλις 100 επαναλήψεις η ακρίβεια και η απώλεια βρίσκονται στο 93% και 0.25 αντίστοιχα ενώ με την ολοκλήρωση όλων των εποχών σε κάθε περίπτωση φτάνουν τις εξής τιμές:

# Εποχών	Ακρίβεια	Απώλεια	Χρόνος
1	98.37%	0.0469	04:00 λεπτά
3	98.73%	0.0393	01:25 λεπτά

Αν και το μοντέλο των 3 εποχών έχει καλύτερα θεωρητικά αποτελέσματα, ο χρόνος που απαιτείται για να εκπαιδευτεί είναι πολύ μεγαλύτερος, το οποίο σημαίνει πως το μοντέλο 1 εποχής είναι μια καλή εναλλακτική. Η διαφορά μεταξύ τους είναι ιδιαίτερα μικρή και σε ακρίβεια αλλά και σε απώλεια. Και τα δύο μοντέλα συγκλίνουν σε επιτυχής προσέγγιση του μη γραμμικού προβλήματος ενώ η έλλειψη κάποιου βασικού trade off μεταξύ των τιμών εκπαίδευσης και ελέγχου υποδηλώνει πως το μοντέλο δεν υπερεκπαιδεύεται στα δεδομένα εκπαίδευσης.

2.1.4. Ερώτημα 4: Confusion Matrix

Για καλύτερη απεικόνιση των συστάδων επιτυχιών αποτυχιών των δύο μοντέλων, κατασκευάστηκε συνάρτηση Matlab που υπολογίζει το Confusion Matrix για κάθε εκπαιδευμένο μοντέλο και τα οποία εμφανίζονται παρακάτω. Βλέπουμε πως οι διαφορές μεταξύ των True Negatives και True Positives μεταξύ των δυο μοντέλων είναι πολύ μικρές σε σχέση με το μέγεθος όλων των δειγμάτων, κάτι που γίνεται και πιο αισθητό αν κάποιος κοιτάξει τα False Positives και False Negatives που στις περισσότερες περιπτώσεις είναι επιπέδου μονάδων.

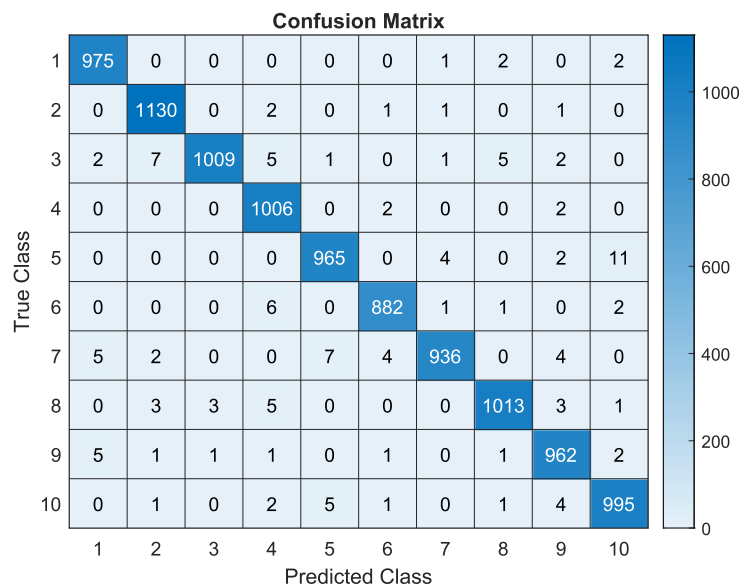
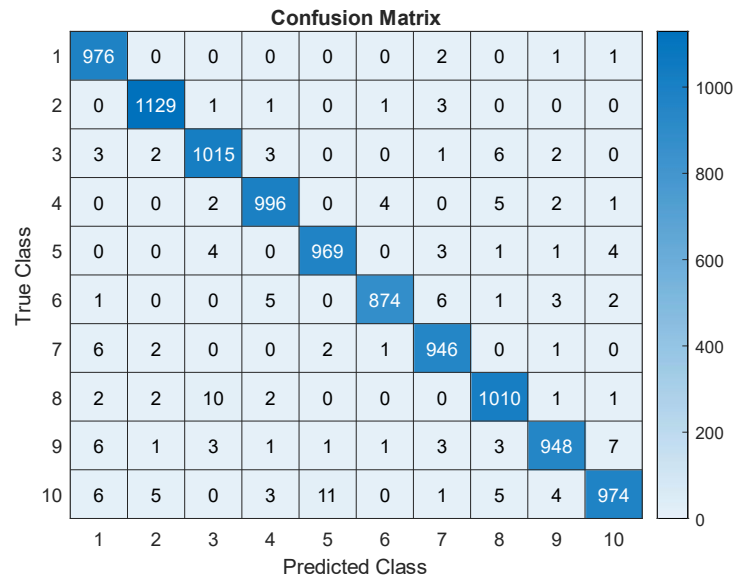


Figure 1: 3 Εποχές – Confusion Matrix



Εικόνα 6: 1 Εποχή - Confusion Matrix

2.2. HOG – Support Vector Machines

Ένας άλλος τρόπος κατηγοριοποίησης των εικόνων είναι η χρήση περιγραφέα χαρακτηριστικών HOG για την ανίχνευση των διαφόρων κλίσεων εσωτερικά της εικόνας σε υποπεριοχές. Συγκεκριμένα χρησιμοποιώντας τα Gradients ανά κατεύθυνση του δισδιάστατου χώρου καθώς και την κλίση (orientation) και το μέγεθος (magnitude) των gradients για υποπεριοχές της εικόνας κατά 8x8 παράθυρα δημιουργούνται διανύσματα histogram βασισμένα σε bins με όρια ανά μοίρες στα οποία προστίθενται η αντίστοιχη τιμή magnitude ανάλογα με την τιμή της κλίσης του αντίστοιχου σημείου. Το ιστογράμμο αυτό έπειτα ενώνεται σε μεγαλύτερο διάνυσμα μεγέθους 36x1 που περιλαμβάνει συνολικά ιστογράμματα 4^{ων} 8x8 περιοχών, κανονικοποιείται με την 2^η νόρμα (μέτρο διανύσματος) για κάθε στοιχείο. Η ένωση όλων των διανυσμάτων που προκύπτουν είναι και το τελικό διάνυσμα του περιγραφέα[1]. Για την σωστή υλοποίηση του χρησιμοποιήθηκαν η έτοιμη συνάρτηση της Matlab ενώ κατασκευάστηκε συνάρτηση στην οποία υλοποιείται και «χειροκίνητα», όπου χρησιμοποιήθηκαν διάφορες πηγές όπως [10] [11].

Χρησιμοποιώντας τον περιγραφέα αυτόν σαν είσοδο για ένα SVM προκύπτει καλύτερη διαχώριση του πολυδιάστατου χώρου με πιο ακριβής εκπαίδευση πολλαπλών SVM για κατηγοριοποίηση περισσότερων από δυο κλάσεων. Τα SVM χρησιμοποιούν διανύσματα υποστηρίξης για την μεγιστοποίηση της απόστασης μεταξύ των υπερεπιπέδων των κλάσεων, με τα καταλυτικά διάνυσμα εκπαίδευσης να παίρνουν τον ρόλο διανυσμάτων υποστηρίξης των διαφόρων υπερεπιπέδων[1]. Η χρήση του περιγραφέα κατευθύνει την εκπαίδευση των SVM με επιπλέον διαστατική πληροφορία της δομής και του σχήματος του αντικειμένου δίνοντας έτσι στα SVM επιπλέον υπερεπίπεδα διαχωρισμού του πολυδιάστατου χώρου εισόδου[12].

2.2.1. Ερώτημα 1: In-Built HOG and SVM

Έγινε χρήση της built in HOG συνάρτησης για κάθε μία από τις εικόνες του σετ με αποτέλεσμα την δημιουργία ενός μητρώου όλων των διανυσμάτων του περιγραφέα. Η ακρίβεια του κατηγοριοποιητή είναι 97.63%, μέσος όρος 5 runs, ενώ η μορφή του φαίνεται στην παρακάτω εικόνα:

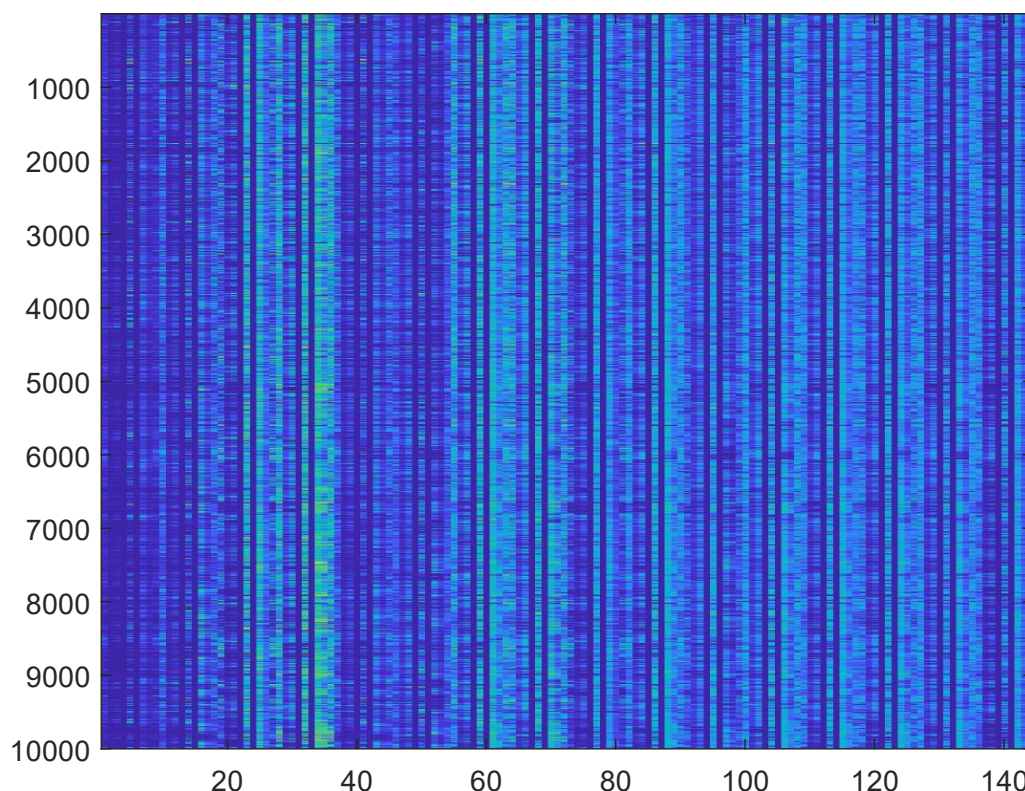

```

svmClassifier =
  ClassificationECOC
    ResponseName: 'Y'
    CategoricalPredictors: []
    ClassNames: [0 1 2 3 4 5 6 7 8 9]
    ScoreTransform: 'none'
    BinaryLearners: {45x1 cell}
    CodingName: 'onevsone'

```

Εικόνα 7: Κατηγοριοποιητής ECOC με πολλαπλά SVM.

Για καλύτερη διαίσθηση του περιγραφέα δίνεται επίσης και το μητρώο όλων των διανυσμάτων του στο παρακάτω γραφικό στοιχείο. Κάθε γραμμή είναι και ένα διάνυσμα περιγραφής για τις εικόνες του σετ δοκιμών:



Εικόνα 8: Μητρώο περιγραφών HOG για το σετ δοκιμών

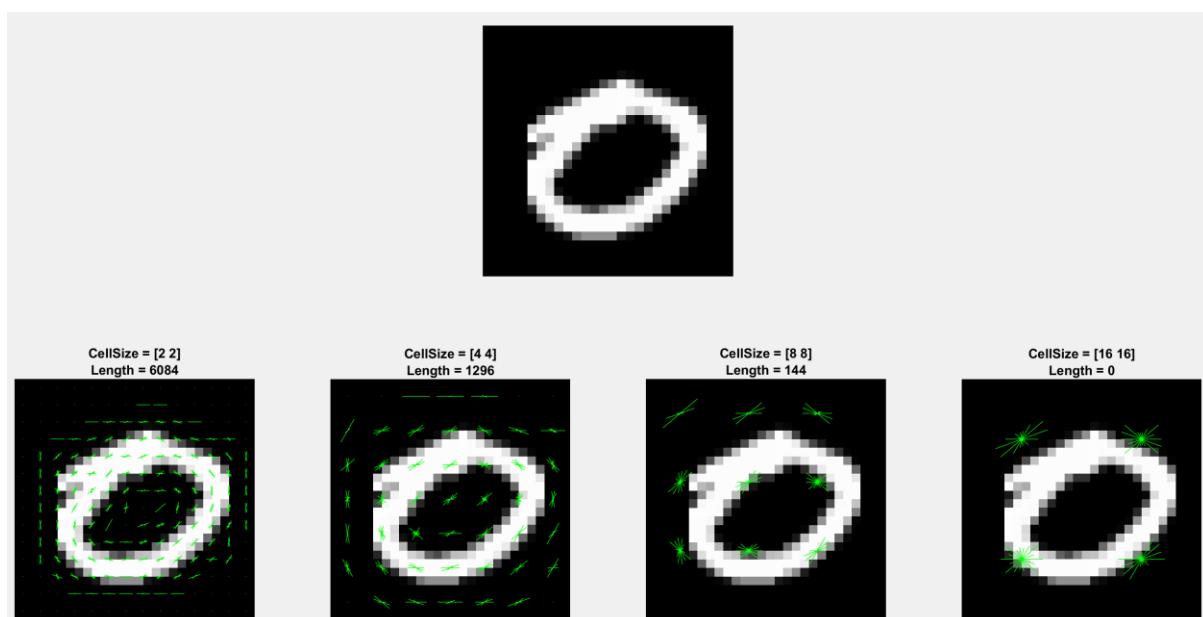
2.2.2. Ερώτημα 2: Αποτελέσματα – HOG Patches

Τα αποτελέσματα του κατηγοριοποιητή είναι πολύ ικανοποιητικά, ο πολυεπίπεδος χώρος φαίνεται να χωρίζεται σε μεγάλο βαθμό από τα υπερεπίπεδα χωρίς πολλά λάθη. Παραδειγματικά παραδίδονται τυχαία οι ετικέτες 10 στοιχείων του συνόλου ελέγχου μαζί με τα αποτελέσματα της κατηγοριοποίησης τους καθώς και τα διανύσματα NegLOSS του κατηγοριοποιητή. Είναι εμφανές ότι ακόμα και σε παραδείγματα μικρογραφίας του συνολικού σετ εικόνων δοκιμής υπάρχει καλό ποσοστό επιτυχίας. Παράλληλα έγινε χρήση διαφορετικών μεγεθών παραθύρου patch με την σύγκριση των αποτελεσμάτων να εμφανίζεται παρακάτω:

TrueLabel	PredLabel	10x10 single matrix									
0	2	-0.1763	-0.1274	-0.0766	-0.1132	-0.2590	-0.3115	-0.4141	0	-0.1044	-0.0427
1	1	-0.0806	0	-0.1690	-0.3130	-0.0507	-0.3271	-0.1156	-0.1627	-0.1122	-0.1324
1	1	-0.1744	-0.2815	-0.2265	-0.0916	-0.0490	-0.1912	-0.2077	-0.0751	-0.0636	0
0	0	-0.1065	0	-0.2077	-0.2503	-0.0706	-0.2074	-0.1144	-0.1539	-0.0748	-0.1091
9	9	-0.2308	-0.2328	-0.2375	-0.0928	-0.1144	-0.1393	-0.3049	-0.0304	-0.0527	0
5	5	-0.1609	-0.3374	-0.2223	-0.0857	-0.0762	-0.1496	-0.2840	-0.0705	-0.0940	0
3	3	-0.2366	-0.1303	-0.1974	-0.0975	0	-0.1627	-0.1507	-0.1649	-0.1311	-0.0539
1	1	-0.2728	-0.2032	-0.0431	0	-0.3089	-0.2034	-0.2343	-0.1344	-0.0847	-0.2308
2	2	-0.1385	-0.1914	0	-0.0575	-0.1402	-0.1847	-0.3868	-0.1686	-0.0490	-0.1202
2	2	-0.0421	-0.0001	-0.1333	-0.2237	-0.0919	-0.2352	-0.0468	-0.2768	-0.0650	-0.1855

Εικόνα 9: 10 τυχαίες ετικέτες και η κατηγοριοποίηση τους μαζί με τα διανύσματα NegLoss

Για καλύτερη κατανόηση των διαφορετικών μεγεθών patches γίνεται εμφάνιση των διανυσμάτων περιγραφής επάνω στην αρχική εικόνα:



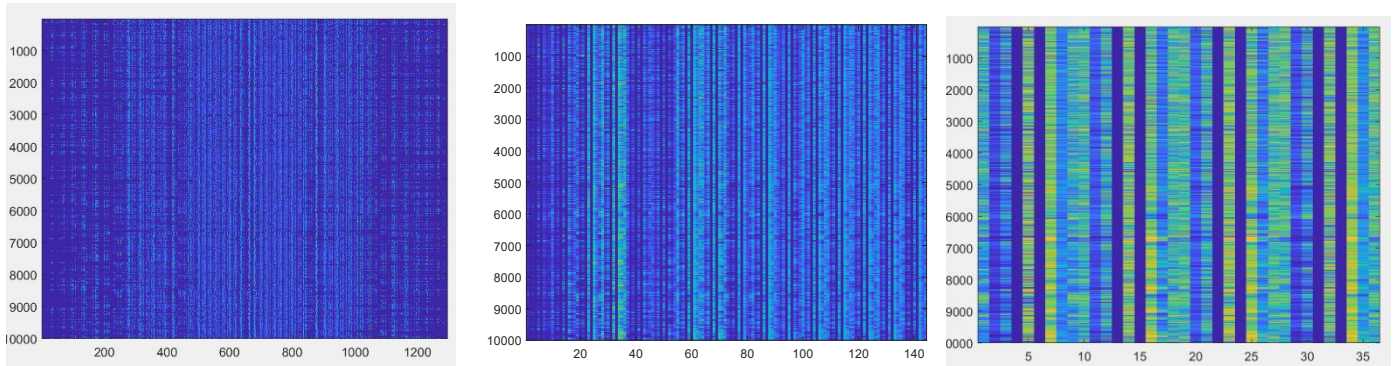
Εικόνα 10: Παράδειγμα παραθύρων HOG με προβολή επάνω στην εικόνα

Τα αναλυτικά αποτελέσματα όλων των διαφορετικών patches που χρησιμοποιήθηκαν εμφανίζονται στον παρακάτω πίνακα:

Μέγεθος Παραθύρου	Ακρίβεια SVM
2x2	Out of Memory
4x4	98.40%
8x8	97.63%
14x14	86.66%

Προφανής παρατήρηση είναι πως όσο μικρότερο παράθυρο χρησιμοποιήσουμε τόσο μεγαλύτερη χωρική πληροφορία των σχημάτων και των κλίσεων δεχόμαστε και αυτομάτως τα αποτελέσματα της εκπαίδευσης είναι καλύτερα. Η διαφορά είναι όμως αρκετά μικρή μεταξύ των 4x4 και 8x8 με το πρώτο παράθυρο να έχει χωρική απαίτηση μόνον για το σετ εκπαίδευσης 60000x1296 ενώ το δεύτερο παράθυρο απαίτηση 60000x144. Η διαφορά της χωρικής πολυπλοκότητας μεταξύ τους για το ελάχιστο κέρδος ακρίβειας που επιφέρεται δεν είναι ρεαλιστική απαίτηση για καθημερινές εφαρμογές. Το μεγαλύτερο παράθυρο από 14x14 έχει σημαντική απώλεια και δεν συνιστάται.

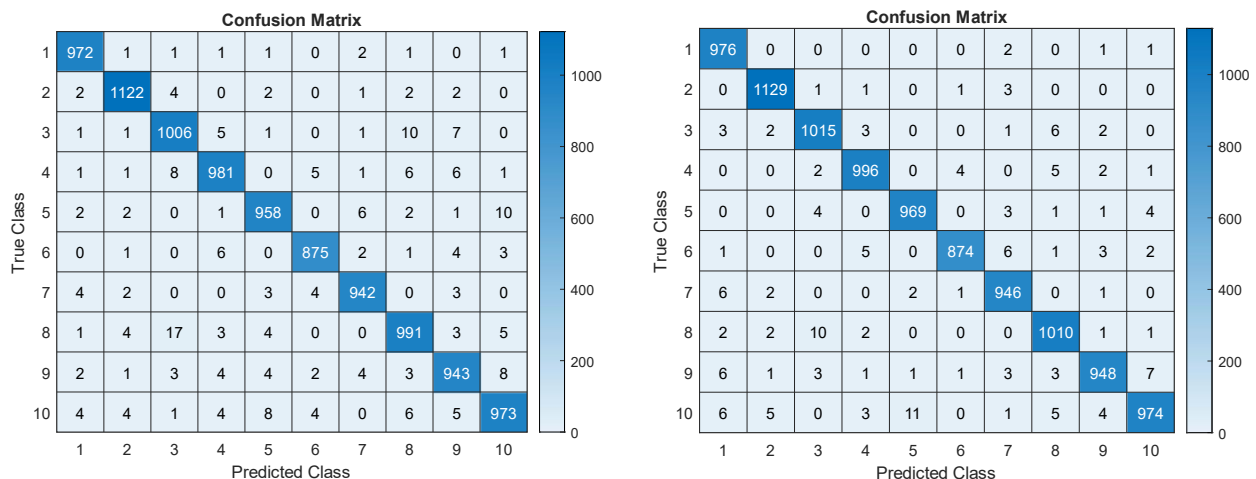
Τα HOG μητρώα για τα τρία από τα 4 patches:



Εικόνα 11: Τα μητρώα HOG για το σετ δοκιμής για 4x4, 8x8 και 14x14 μεγέθη παραθύρου

2.2.3. Ερώτημα 3: Confusion Matrix

Για καλύτερη απεικόνιση των επιτυχιών αποτυχιών μεταξύ των δύο μοντέλων HOG-SVM 8x8 και CNN εμφανίζεται παρακάτω το Confusion Matrix και των δύο μεθόδων για λόγους σύγκρισης. Χρησιμοποιήθηκε η Custom συνάρτηση που κατασκευάστηκε για το Confusion Matrix για κάθε εκπαιδευμένο μοντέλο. Βλέπουμε πως οι διαφορές μεταξύ των True Negatives/True Positives καθώς και False Positives/False Negatives μεταξύ των δυο μοντέλων είναι μικρές καθώς και τα δύο μοντέλα έχουν καταφέρει να συγκρίνουν σε πολύ μεγάλο επίπεδο ακρίβειας. Παρόλαυτα εμφανίζονται κάποιες διαφορές.



Εικόνα 12: Τα Confusion μητρώα για το HOG-SVM 8x8 και CNN αντίστοιχα

Οι κυριότερες διαφορές είναι πως για όλες τις κλάσεις εκτός του ψηφίου 6 το CNN έχει περισσότερα True Positives με τα μεγαλύτερα λάθη να συνέβησαν στο SVM για τις κλάσεις 8-3 και 3-8 καθώς και 5-10 (true-predicted) ενώ για το CNN το 8-3 και 10-5. Υπενθυμίζουμε πως οι ακρίβειες των δύο μοντέλων είναι 97.63%(SVM) και 98.73%(CNN – 3 epochs) αντίστοιχα, με το καλύτερο HOG παράθυρο να προσφέρει 98.40%.

2.2.4. Ερώτημα 4: Bonus: Custom HOG

Για την υλοποίηση του Custom HOG κατασκευάστηκε συνάρτηση με την βοήθεια διαφόρων πηγών όπως [10] [11]. Δίνεται ο κώδικας παρακάτω με πλήρη επεξήγηση σε μορφή σχολίων:

```
% Function does not output right results. Error in bin and cell_hist!!

function features = customHOG(image, cellSize, blockSize, numBins)
% Calculate gradients using Sobel filters
[dx, dy] = imgradientxy(image, 'Sobel');

% Calculate gradient magnitude and orientation
magnitude = sqrt(dx.^2 + dy.^2);
orientation = atan2(dy, dx);

% Determine the size of the HOG grid
[imageHeight, imageWidth] = size(image);

% Initialize the HOG feature vector
features = [];

block = blockSize(1) * cellSize(1);
jump = size(image,1) - block;

%% Iterate through blocks and compute HOG features with block interleaving
for y = 1:jump:2*jump
    for x = 1:jump:2*jump
        % Initialize block histogram
        blockHistogram = [];

        % Iterate through cells within the block
        for by = 0:1
            for bx = 0:1
                % Define cell boundaries
                xStart = x + bx * cellSize(1);
                xEnd = xStart + cellSize(1) - 1;
                yStart = y + by * cellSize(2);
                yEnd = yStart + cellSize(2) - 1;

                % Select gradients and orientations within the cell
                cellMagnitude = magnitude(yStart:yEnd, xStart:xEnd);
                cellOrientation = orientation(yStart:yEnd, xStart:xEnd);

                cell_hist = zeros(1, numBins);

                for k = 1:8
                    for l = 1:8

                        % Determine the bin index
                        bin = floor((cellOrientation(k,l) / 180) * numBins) + 1;
                        bin = max(min(bin, numBins), 1);

                        % Add magnitude to the appropriate bin
                        cell_hist(bin) = cell_hist(bin) + cellMagnitude(k,l);

                    end
                end

                % Concatenate the block histogram to the feature vector
                blockHistogram = [blockHistogram cell_hist];

            end
        end

        % L2 normalization within each block
        blockNorm = norm(blockHistogram);
        blockFeatures = blockHistogram / blockNorm;
        features = [features blockFeatures];
    end
end

%% Function to create the cell Histogram
function cellHistogram = computeCellHistogram(cellMagnitude, cellOrientation, numBins)
cellHistogram = zeros(1, numBins);
for bin = 1:numBins
    binRange = [(bin - 1) * (180 / numBins), bin * (180 / numBins)];
    binMask = (cellOrientation >= binRange(1)) & (cellOrientation < binRange(2));
    cellHistogram(bin) = sum(cellMagnitude(binMask));
end
end
```

3. Κώδικες Υλοποίησης

[Σελίδα GitHub](#)

References

- [1] ‘Εργαστηριακή Άσκηση - Μέρος Β 2022-2023.pdf’.
- [2] ‘MNIST’ handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges’. <http://yann.lecun.com/exdb/mnist/> (accessed Sep. 21, 2023).
- [3] M. Mayer, ‘MNIST database of handwritten digits in MATLAB format’. Jul. 16, 2023. Accessed: Sep. 21, 2023. [Online]. Available: <https://github.com/sunsided/mnist-matlab>
- [4] ‘Read digits and labels from MNIST database’, Sep. 21, 2023. <https://www.mathworks.com/matlabcentral/fileexchange/27675-read-digits-and-labels-from-mnist-database> (accessed Sep. 21, 2023).
- [5] S. Jaeger, ‘The Golden Ratio of Learning and Momentum’, 2020, doi: 10.48550/ARXIV.2006.04751.
- [6] ‘Batch normalization in 3 levels of understanding | by Johann Huber | Towards Data Science’. <https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338#b93c> (accessed Sep. 21, 2023).
- [7] ‘Classification of MNIST Data using CNN - File Exchange - MATLAB Central’. <https://www.mathworks.com/matlabcentral/fileexchange/103575-classification-of-mnist-data-using-cnn> (accessed Sep. 21, 2023).
- [8] ‘Create Simple Deep Learning Neural Network for Classification - MATLAB & Simulink Example’. <https://www.mathworks.com/help/deeplearning/ug/create-simple-deep-learning-network-for-classification.html> (accessed Sep. 21, 2023).
- [9] R. C. Gonzalez, ‘Deep Convolutional Neural Networks [Lecture Notes]’, *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 79–87, Nov. 2018, doi: 10.1109/MSP.2018.2842646.
- [10] ‘Digit Classification Using HOG Features - MATLAB & Simulink’. <https://www.mathworks.com/help/vision/ug/digit-classification-using-hog-features.html> (accessed Sep. 22, 2023).
- [11] ‘Histogram of Oriented Gradients (HOG) code using Matlab - File Exchange - MATLAB Central’. https://www.mathworks.com/matlabcentral/fileexchange/46408-histogram-of-oriented-gradients-hog-code-using-matlab?s_tid=FX_rc3_behav (accessed Sep. 22, 2023).
- [12] A. J. Smola and B. Schölkopf, ‘A tutorial on support vector regression’, *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004, doi: 10.1023/B:STCO.0000035301.49549.88.