Πανεπιστήμιο Πατρών
Τμήμα Μηχανικών Η/Υ και Πληροφορικής

# Ψηφιακή Επεξεργασία και Ανάλυση Εικόνας

## Εργαστηριακή Άσκηση - Μέρος Β

Ακαδημαϊκό Έτος 2023/24

Ζήσης Σούρλας *

17 Σεπτεμβρίου 2024

*AM: 1072477 Email: sourlas.zisis@ac.upatras.gr

# Περιεχόμενα

# 1 Απεικόνιση δειγμάτων

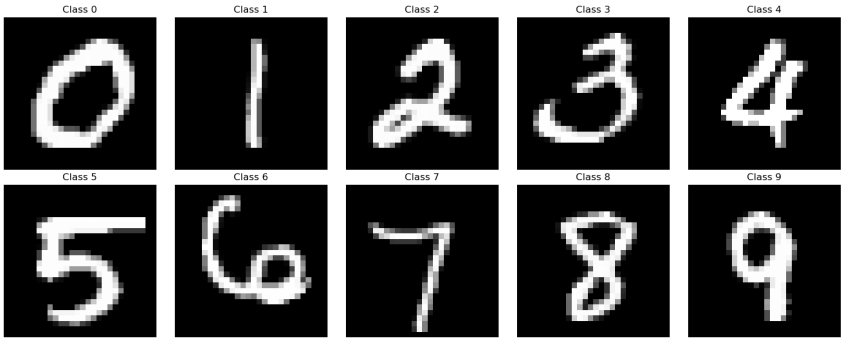



Figure 1: Δείγματα από κάθε κλάση του MNIST

# 2 Διαχωρισμός συνόλου δεδομένων

Το σύνολο δεδομένων χωρίστηκε σε τρία μέρη. Στο υποσύνολο εκπαίδευσης (train), ελέγχου (test) και επικύρωσης (validation). Ως υποσύνολο επικύρωσης χρησιμοποιήθηκε το παρεχόμενο test set του MNIST ενώ για τα άλλα δύο σύνολα χρησιμοποιήθηκε το παρεχόμενο train set διαχωρισμένο σε δύο μη επικαλυπτόμενα μέρη, με αναλογία 80%-20% για τα υποσύνολα εκπαίδευσης και επικύρωσης αντίστοιχα.

Για την λήψη του συνόλου δεδομένων χρησιμοποιήθηκε η ενσωματωμένη κλάση MNIST της βιβλιοθήκης **torchvision**. Για τον διαχωρισμό σε υποσύνολα χρησιμοποιήθηκε η συνάρτηση **random_split** της βιβλιοθήκης **pytorch**. Επιπλέον, για τη φόρτωση των δεδομένων στο νευρωνικό δίκτυο σε μη επικαλυπτόμενα υποσύνολα (batches) χρησιμοποιήθηκε η κλάση **dataloader** της **pytorch**.

# 3 Διαδικασία εκπαίδευσης και αξιολόγησης

## 3.1 Εκπαίδευση

Για την εκπαίδευση των νευρωνικών δικτύων (τόσο του CNN όσο και του απλού FC Classifier) δημιουργήθηκε η συνάρτηση **nn_train**. Αυτή δέχεται ως είσοδο το μοντέλο (ως pytorch module) που θα εκπαιδευτεί, τους dataloaders των υποσυνόλων εκπαίδευσης και ελέγχου, τον optimizer (δηλαδή τη συνάρτηση ανανέωσης των βαρών του δικτύου) που θα χρησιμοποιηθεί, τη συνάρτηση απώλειας (loss function) για τον υπολογισμό του σφάλματος στην έξοδο του δικτύου και τον αριθμό των εποχών εκπαίδευσης που θα εκτελεστούν.

Για κάθε εποχή εκπαίδευσης η συνάρτηση φορτώνει τα batches (μεγέθους 256) του συνόλου εκπαίδευσης και για κάθε batch:

1. Υπολογίζει την έξοδο του νευρωνικού δικτύου.
2. Τροφοδοτώντας τη συνάρτηση απώλειας με την έξοδο του δικτύου και με την επιθυμητή έξοδο (το label της αντίστοιχης εικόνας), υπολογίζει το σφάλμα του δικτύου. Το σφάλμα αποθηκεύεται ώστε να υπολογιστεί μετά το μέσο σφάλμα εκπαίδευης.
3. Στη συνέχεια κάνει πίσω διάδωση του σφάλματος (back probagation) ώστε να υπολογιστεί το διάνυσμα κλίσης (gradient).
4. Τέλος χρησιμοποιώντας τον optimizer, κάνει ανανέωση των βαρών βάσει του gradient που υπολογίστηκε.

Στη συνέχεια η συνάρτηση φορτώνει τα batches του συνόλου ελέγχου για τα οποία επαλαμβάνει την παραπάνω διαδικασία χωρίς όμως αυτή τη φορά να υπολογίζει το διάνυσμα κλίσης και χωρίς να κάνει ανανέωση των βαρών. Πέραν της συνάρτησης απώλειας, υπολογίζει το label που προκύπτει από την έξοδο του

δικτύου παίρνοντας τον δείκτη του μεγαλύτερου αριθμού στην έξοδο (δηλαδή την έξοδο με τη μεγαλύτερη ενέργεια / πιθανότητα) και υπολογίζει τον αριθμό των σωστών εξόδων. Τέλος, υπολογίζει την μέση απώλεια και την ακρίβεια του δικτύου στο σύνολο ελέγχου. Αν η μέση απώλεια στο σύνολο ελέγχου είναι η μικρότερη μέχρι στιγμής, το μοντέλο αποθηκεύεται ως το προς ώρας καλύτερο.

### 3.2 Αξιολόγηση

Για την αξιολόγηση του μοντέλου, δημιουργήθηκε η συνάρτηση **nn_eval**. Δέχεται ως είσοδο το μοντέλο που θα αξιολογηθεί, τον dataloader του συνόλου αξιολόγησης και την συνάρτηση απώλειας που θα χρησιμοποιηθεί. Επί της ουσίας, εκτελεί ότι και η **nn_train** για το σύνολο ελέγχου, διατηρώντας και αποθηκεύοντας ωστόσο τις προβλέψεις του μοντέλου και τις επιθυμητές εξόδους. Αυτές χρησιμοποιούνται στη συνέχεια για την παραγωγή του μητρώου σύγχυσης.

## 4 Υπολογισμός μητρώου σύγχυσης

Για τον υπολογισμό του μητρώου σύγχυσης χρησιμοποιήθηκε η συνάρτηση **confusion_matrix** της **sklearn**, ενώ για την απεικόνισή του η συνάρτηση **heatmap** της βιβλιοθήκης **seaborn**. Στην πρώτη δίνονται ως είσοδοι οι προβλέψεις του μοντέλου και οι πραγματικές ετικέτες και αυτή παράγει το μητρώο σύγχυσης. Στη συνέχεια αυτό δίνεται ως είσοδος στη δεύτερη συνάρτηση η οποία παράγει την απεικόνιση.

## 5 Εξαγωγή χαρακτηριστικών

### 5.1 Συνελεκτικό Νευρωνικό Δίκτυο (CNN)

#### 5.1.1 Θεωρητικό υπόβαθρο

Τα συνελικτικά νευρωνικά δίκτυα είναι μια κατηγορία νευρωνικών δικτύων που χρησιμοποιούνται πολύ συχνά για την εξαγωγή χαρακτηριστικών σε εικόνες. Η συνελικτική τους ιδιότητα, τους επιτρέπει να εξάγουν χαρακτηριστικά αυτομάτως, απλώς με την παροχή μεγάλων συνόλων δεδομένων που ανταποκρίνονται σε μια συγκεκριμένη εργασία χωρίς να χρειάζεται χειροκίνητη κατασκευή χαρακτηριστικών.

Τα βασικά δομικά συστατικά των CNNs είναι:

1. Συνελικτικά στρώματα (convolutional layers)
2. Στρώματα συνένωσης (pooling layers)
3. Ένα πλήρως συνδεδεμένο στρώμα (fully connected layer)

Τα συνελικτικά στρώματα χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών. Κάθε στρώμα αποτελείται από μία σειρά από πυρήνες (kernels) (6 και 12 στην περίπτωσή μας) συγκεκριμένου μεγέθους παραθύρου (2 επί 2 εν προκειμένω). Οι πυρήνες αυτοί διατρέχουν όλη την εικόνα σε διαδοχικά μπλοκ (η απόσταση τους καθορίζεται από το stride) και για κάθε μπλοκ εικονοστοιχείων υπολογίζεται το άθροισμα των τιμών των εικονοστοιχείων επί τα αντίστοιχα βάρη του κάθε πυρήνα, το οποίο στη συνέχεια τροφοδοτείται σε μία συνάρτηση ενεργοποίησης (εδώ ReLU), η έξοδος της οποία είναι και η έξοδος του στρώματος. Τα προαναφερθέντα βάρη προσαρμόζονται κατά την εκπαίδευση ώστε να παράγουν χαρακτηριστικά που καθιστούν τις εικόνες ευκολότερα διαχωρίσιμες βάσει της εργασίας που θέλουμε να επιτελεί το δίκτυο.

Τα pooling layers χρησιμοποιούνται για τη μείωση της διαστατικότητας των χαρακτηριστικών. Σκοπός είναι να διατηρηθούν μόνο τα σημαντικότερα χαρακτηριστικά ώστε να επιταχυνθεί η επεξεργασία των δεδομένων. Ωστόσο, η μείωση της διαστατικότητας σημαίνει παράλληλα και απώλεια πληροφορίας και γι' αυτό πρέπει να χρησιμοποιείται καταλλήλως. Αυτά τα στρώματα λειτουργούν ως εξής: Χωρίζουν την εικόνα σε μπλοκ (εδώ 2 επί 2) και αντικαθιστούν κάθε μπλοκ με μία τιμή. Αυτή μπορεί να είναι η μέγιστη τιμή του μπλοκ (όπως γίνεται και στην παρούσα εργασία), η μέση τιμή του μπλοκ ή η ευκλείδεια νόρμα του.

Τέλος, το πλήρως συνδεδεμένο στρώμα χρησιμοποιείται για την ταξινόμηση των δειγμάτων. Λαμβάνει τα χαρακτηριστικά που εξάγει το τελευταίο στρώμα του συνελικτικού νευρωνικού δικτύου ως διάνυσμα πλέον, και δίνει ως έξοδο ένα διάνυσμα μήκους ίσο με τον αριθμό των κλάσεων του προβλήματός μας. Κάθε τιμή του διανύσματος αντιστοιχεί στην ενέργεια της αντίστοιχης κλάσης για τη δοσμένη είσοδο. Η κλάση με τη μεγαλύτερη ενέργεια θεωρείται και η κλάση στην οποία ταξινομείται η είσοδος. Αν στην έξοδο του εφαρμοστεί μια softmax συνάρτηση, τότε επιστρέφεται η πιθανότητα με την οποία τον δοθέν δείγμα αντιστοιχίζεται σε κάθε κλάση.

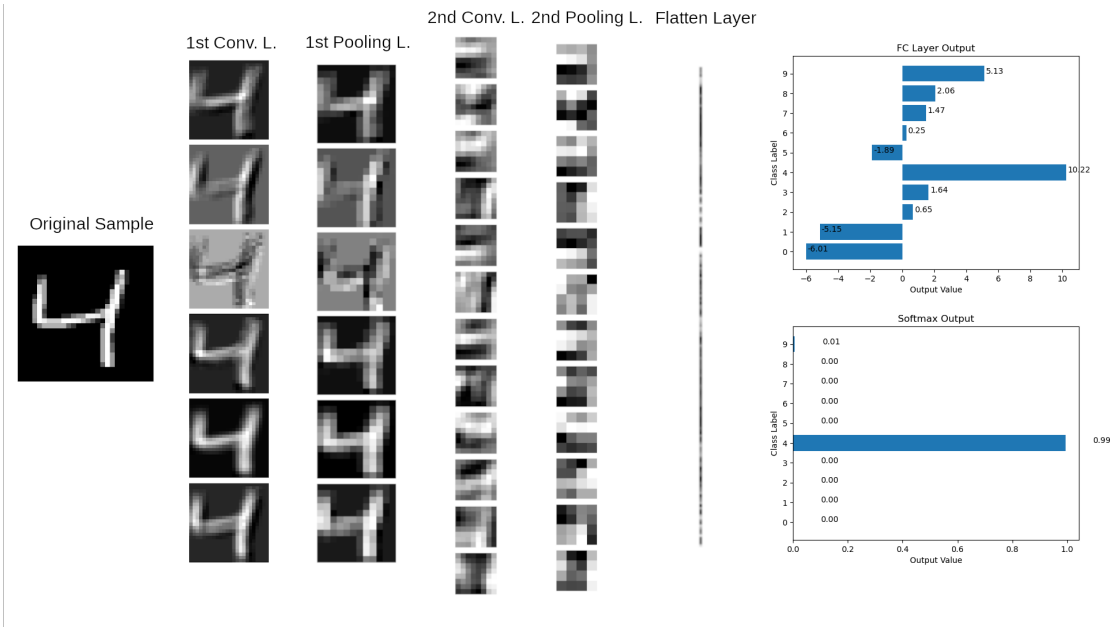



Figure 2: Οι έξοδοι των στρωμάτων του εκπαιδευμένου ΣΝΔ για μεμονομένο δείγμα

### 5.1.2 Υλοποίηση

Για την υλοποίηση του Συνελικτικού Νευρωνικού Δικτύου, δημιουργήθηκε η κλάση **CNNClassifier** που κληρονομεί από την κλάση **nn.Module** της pytorch. Στη μέθοδο __**init**__ ορίζονται όλα τα στρώματα που χρησιμοποιεί το δίκτυο:

- Δύο συνελικτικά στρώματα δύο διαστάσεων, σύμφωνα με τις προδιαγραφές της εκφώνησης, αξιοποιώντας την κλάση **nn.Conv2d** της pytorch.
- Η συνάρτηση ενεργοποίησης αξιοποιώντας την κλάση **nn.ReLU**.
- Το pooling layer αξιοποιώντας την κλάση **nn.MaxPool2d**.
- Ένα στρώματα για την μετατροπή της εισόδου σε διάνυσμα (**nn.Flatten**).
- Και το πλήρως συνδεδεμένο στρώμα χρησιμοποιώντας τη συνάρτηση **nn.Linear**.

Στη μέθοδο **forward** ορίζεται η σειρά με την οποία η είσοδος διέρχεται από τα διάφορα στρώματα του δικτύου και η οποία ακολουθεί την περιγραφή της εκφώνησης.

Για την εκπαίδευση και την αξιολόγηση του δικτύου χρησιμοποιήθηκαν η **nn_train** και η **nn_eval** αντίστοιχα. Ως optimizer χρησιμοποιήθηκε ο SGD (**torch.optim.SGD**) με **ρυθμό μάθησης (learning rate) 0.001 και ορμή (momentum) 0.9**.

Ως συνάρτηση απώλειας χρησιμοποιήθηκε η διεντροπία (Cross Entropy), η οποία ενδείκνυται για προβλήματα ταξινόμησης, αξιοποιώντας την κλάση **nn.CrossEntropyLoss** της pytorch.

### 5.1.3 Πειραματική αξιολόγηση

Μια πρώτη παρατήρηση είναι πως το συνελικτικό νευρωνικό δίκτυο ανταποκρίνεται πολύ καλά στις ανάγκες του προβλήματος μας. Χρησιμοποιώντας ολόκληρο το σύνολο εκπαίδευσης, καταφέρνει να επιτύχει 96% (βλ. 5) ακρίβεια ενώ η εκπαίδευσή του είναι αρκετά γρήγορη και επί της ουσίας ολοκληρώνεται μετά από 10-12 εποχές (βλ. 3). Αυτό μας δείχνει ότι το μοντέλο έχει καταφέρει να εξάγει κατάλληλα χαρακτηριστικά που καθιστούν εύκολα και σαφώς διαχωρίσιμα τα δεδομένα διαφορετικών κλάσεων. Η σύγκλιση του μοντέλου παρατηρείται με βάση το σύνολο ελέγχου καθώς αυτό αποτελείται από δεδομένα που το μοντέλο δεν έχει ξαναδεί και τα οποία δεν χρησιμοποιούνται στην εκπαίδευση. Αυτό σημαίνει πως αν το μοντέλο ταξινομεί σωστά αυτά τα δεδομένα, έχει επιτύχει την ικανότητα να γενικεύει, δηλαδή να ανταποκρίνεται σε άγνωστα δεδομένα του ίδιου προβλήματος.

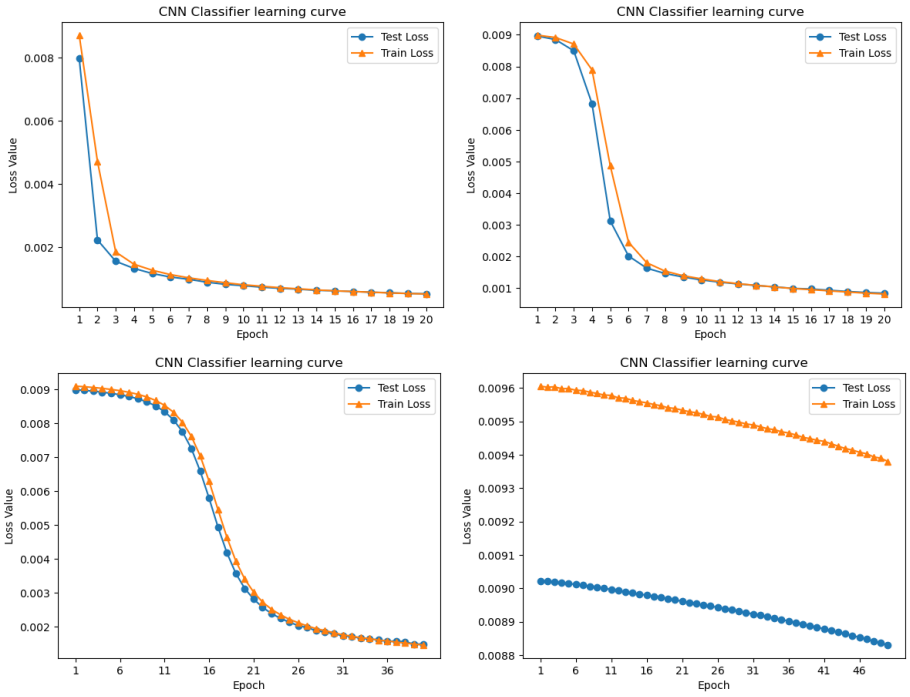



Figure 3: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Παρατηρούμε επιπλέον ότι η μείωση του συνόλου εκπαίδευσης στο 50% και στο 10% δεν επηρεάζει δραματικά τις επιδόσεις του μοντέλου. Η ακρίβεια παραμένει

αρκετά υψηλή ωστόσο η ταχύτητα σύγκλισης στη δεύτερη περίπτωση είναι αρκετά μειωμένη (βλ. 3) καθώς απαιτούνται περίπου 30 εποχές.

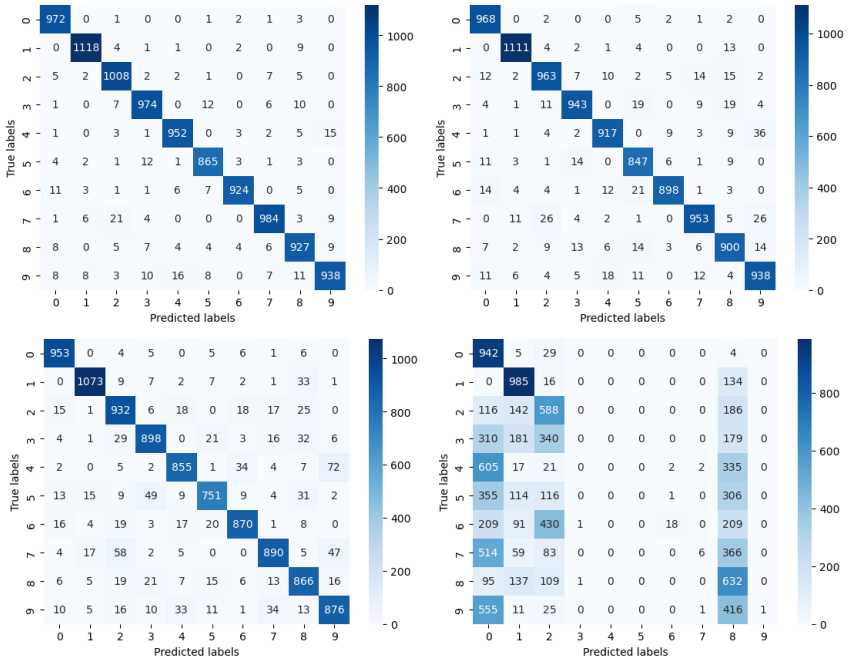



Figure 4: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Στην περίπτωση που χρησιμοποιήσουμε πολύ μικρό σύνολο εκπαίδευσης (1% του αρχικού), το μοντέλο αποτυγχάνει παντελώς να συγκλίνει. Αυτό είναι εμφανές στην καμπύλη σύγκλισης (3) αλλά επιβεβαιώνεται και από την ακρίβεια του μοντέλου που είναι πάρα πολύ χαμηλή και από το μητρώο σύγχυσης που εμφανίζει πάρα πολλές λανθασμένες ταξινομήσεις (βλ. 4).

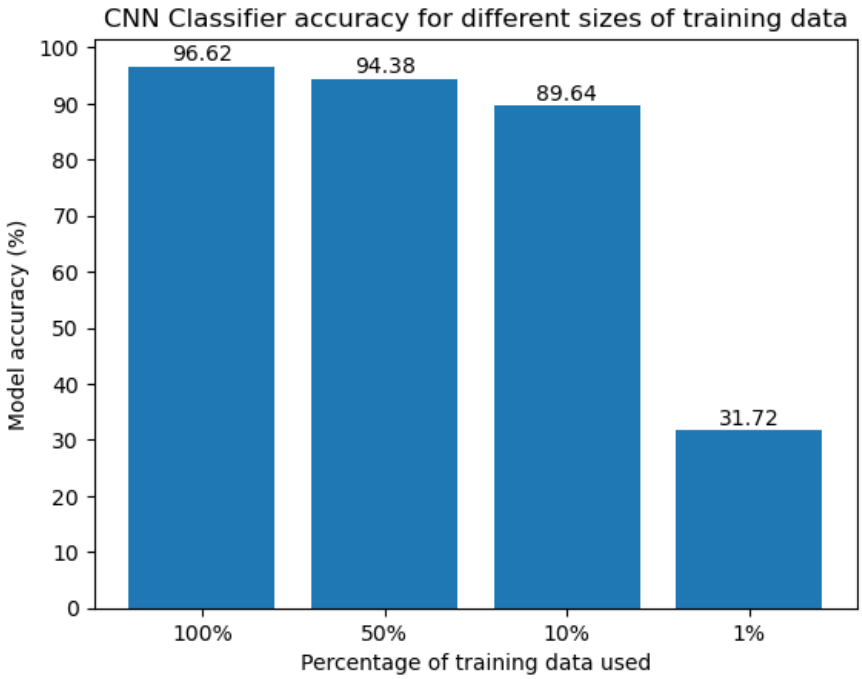



Figure 5: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

## 5.2 Principal Component Analysis (PCA)

### 5.2.1 Θεωρητικό υπόβαθρο

Η Ανάλυση Κυρίων Συνιστωσών είναι μία γραμμική μέθοδος μείωσης της διαστατικότητας των δεδομένων. Βασίζεται στην υπόθεση ότι οι μεταβλητές των δεδομένων εμφανίζουν γραμμικές συσχετίσεις. Στόχος είναι η προβολή τους σε ένα χώρο μικρότερης διάστασης, διατηρώντας ταυτόχρονα όσο το δυνατόν περισσότερη πληροφορία. Αυτό επιτυγχάνεται υπολογίζοντας τα ιδιοδιανύσματα (κύριες συνιστώσες) του μητρώου αυτοσυσχέτισης των δεδομένων. Αυτά τα ιδιοδιανύσματα υποδεικνύουν την κατεύθυνση της διασποράς των δεδομένων. Όσο μεγαλύτερη είναι η ιδιοτιμή που αντιστοιχεί σε ένα ιδιοδιάνυσμα, τόσο μεγαλύτερη είναι και η διασπορά (άρα και η πληροφορία) των δεδομένων στην κατεύθυνση που αυτό ορίζει. Στόχος της PCA είναι η διατήρηση της μέγιστης δυνατής πληροφορίας, και αυτό επιτυγχάνεται με την επιλογή των ιδιοδιανυσμάτων με τις μεγαλύτερες ιδιοτιμές ώστε να αποτελέσουν τη βάση του νέου χώρου χαμηλότερης διάστασης. Τα αρχικά δεδομένα προβάλλονται πάνω στα επιλεχθέντα ιδιοδιανύσματα και έτσι επιτυγχάνεται η μείωση της διαστατικότητας. Η προσπάθεια για μείωση της διασταστικότητας επί της ουσίας ισοδυναμεί με την εξαγωγή χαρακτηριστικών καθώς και στις δύο περιπτώσεις πρόκειται για αναπαράσταση των δεδομένων σε μικρότερο χώρο, διατηρώντας όμως την πληροφορία που τα διαφοροποιεί μεταξύ τους. Συνεπώς, μπορούμε να χρησιμοποιήσουμε την PCA για να εξάγουμε χαρακτηριστικά.

### 5.2.2 Υλοποίηση

Για την υλοποίηση της PCA δημιουργήθηκε η συνάρτηση **pca_fe** η οποία δέχεται ως είσοδο έναν dataloader για τα δεδομένα για τα οποία θέλουμε να κάνουμε εξαγωγή χαρακτηριστικών και ένα αντικείμενο pca (προαιρετικά). Αφού μετατρέψει κάθε εικόνα σε διάνυσμα, εφόσον δεν της έχει δοθεί κάποιο pca αντικείμενο στην είσοδο (φάση εκπαίδευσης), δημιουργεί ένα νέο αντικείμενο της κλάσης PCA της **sklearn** το οποίο εκπαιδεύει στα δεδομένα εισόδου. Το αντικείμενο αυτό έχει παραμετροποιηθεί ώστε να διατηρεί 128 κύριες συνιστώσες. Αφού ολοκληρωθεί αυτή η διαδικασία ή εφόσον δε βρισκόμαστε σε φάση εκπαίδευσης, εξάγονται οι κύριες συνιστώσες (χαρακτηριστικά) των δεδομένων εισόδου και αποθηκεύονται μαζί με τις ετικέτες σε ένα torch dataset, το οποίο επιστρέφεται.

Αφού εξαχθούν τα χαρακτηριστικά και για τα τρία υποσύνολα (εκπαίδευσης, ελέγχου και αξιολόγησης), γίνεται η εκπαίδευση του ταξινομητή.

Ο ταξινομητής που χρησιμοποιήθηκε τόσο για την PCA όσο και για τις υπόλοιπες μεθόδους που ακολουθούν είναι ένας γενικός ταξινομητής που υλοποιήθηκε ως κλάση με όνομα **GenericClassifier**, και αποτελείται από ένα στρώμα με διάσταση εισόδου που δίνεται κατά τη δημιουργία του αντικειμένου, ώστε να προσαρμόζεται στο μέγεθος του διανύσματος χαρακτηριστικών του κάθε μοντέλου, και διάσταση εξόδου 10, όσες δηλαδή και οι κλάσεις του προβλήματος. Για την εκπαίδευσή του χρησιμοποιήθηκε η **nn_train**, δίνοντας όμως ως optimizer τον Adam αντί του

SGD καθώς είναι προσαρμοστικός αλγόριθμος και μπορεί ευκολότερα να προσαρμοστεί στις ανάγκες κάθε μοντέλου.

### 5.2.3 Πειραματική αξιολόγηση

Παρατηρούμε ότι τα χαρακτηριστικά που παράγονται από την PCA είναι υψηλής ποιότητας και κατάλληλα για το πρόβλημά μας καθώς επιτυγχάνεται ακρίβεια πάνω από 90% στο πλήρες σύνολο εκπαίδευσης (βλ. 8). Επιπλέον, η εκπαίδευση είναι αρκετά γρήγορη, με τον ταξινομητή να χρειάζεται περίπου 10 εποχές για να συγκλίνει (βλ. 6).

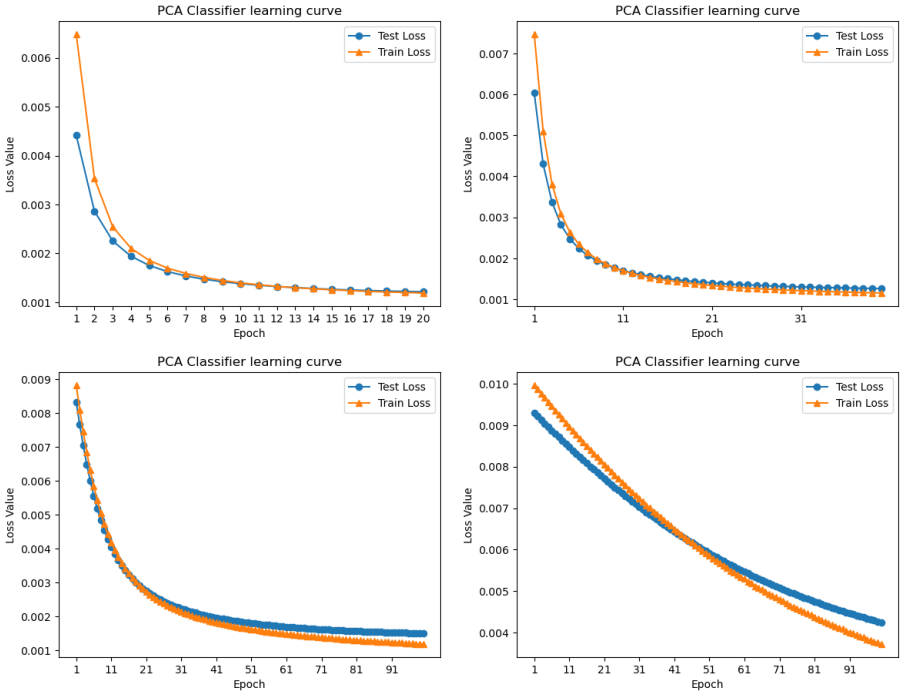



Figure 6: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Επιπλέον, παρατηρούμε ότι η μείωση του συνόλου εκπαίδευσης επηρεάζει πολύ αρνητικά τον χρόνο σύγκλισης της εκπαίδευσης του ταξινομητή εκτοξεύοντάς τον στις 20 περίπου εποχές για το 50% του συνόλου εκπαίδευσης και στις 60 για το 10%. Ωστόσο, η ακρίβεια του μοντέλου παραμένει σχεδόν στάσιμη δείχνοντας την ανθεκτικότητα της μεθόδου σε λίγα δεδομένα.

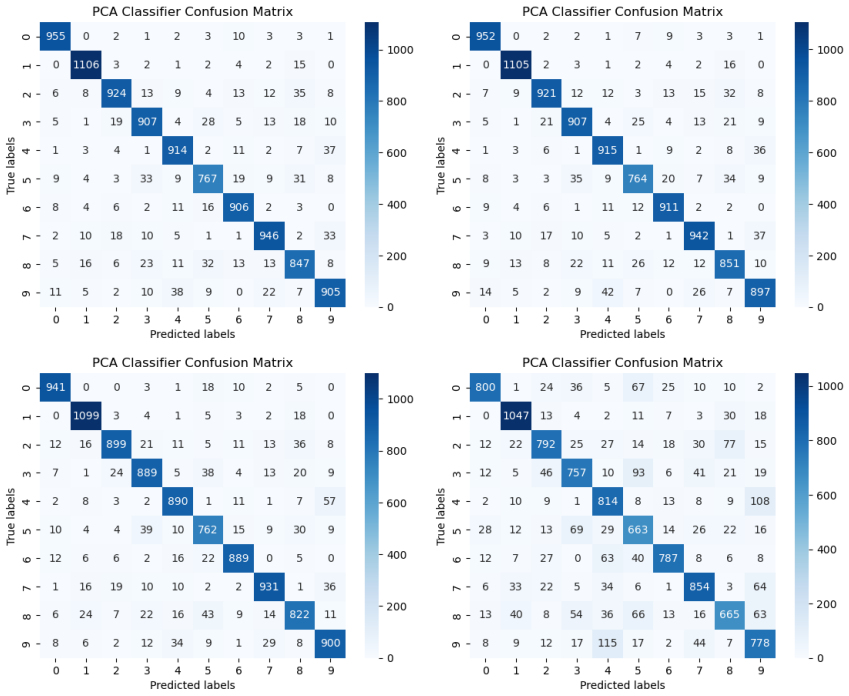



Figure 7: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Η ανθεκτικότητα της PCA καταδεικνύεται επιπλέον από την υψηλή ακρίβεια του μοντέλου ακόμη και όταν χρησιμοποιείται μόλις το 1% του συνόλου εκπαίδευσης. Το μοντέλο επιτυγχάνει σχεδόν 80% ακρίβεια (βλ. 8) παρότι ο ταξινομητής δε δείχνει σημάδια σύγκλισης (βλ. 6). Αυτό υπονοεί ότι αφενός η PCA μπορεί να παράξει καλά χαρακτηριστικά με πολύ μικρό όγκο δεδομένων εκπαίδευσης, γεγονός που οφείλεται στη φύση της ως μεθόδου, και αφετέρου ότι παρά την ελλειπή εκπαίδευσή του ο ταξινομητής είναι ικανός να ταξινομήσει αυτά τα χαρακτηριστικά αρκούντως καλά.

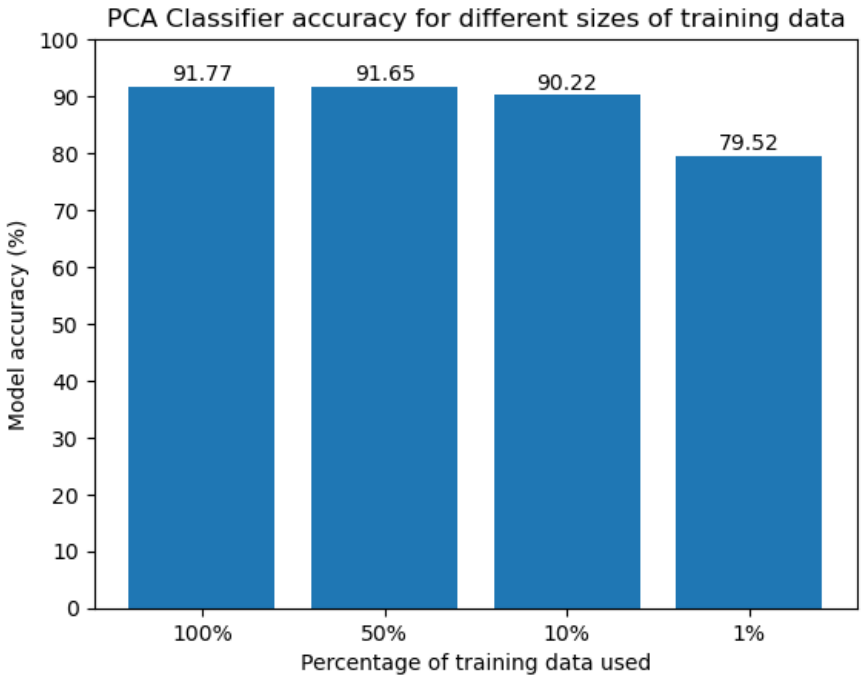



Figure 8: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

## 5.3 Histogram of Oriented Gradients (HOG)

### 5.3.1 Θεωρητικό υπόβαθρο

Το HOG είναι ένας περιγραφέας χαρακτηριστικών (feauture descriptor) που χρησιμοποιείται συχνά σε προβλήματα εντοπισμού αντικειμένων. Η βασική ιδέα στην οποία βασίζεται αυτή η μέθοδος, είναι ότι το σχήμα και το περίγραμμα ενός αντικειμένου μπορούν να περιγραφούν αποτελεσματικά από την κλίση (gradient) της εικόνας, η οποία δείχνει την κατεύθυνση της αλλαγής στην έντασης της εικόνας.

Για την εξαγωγή των χαρακτηριστικών, αρχικά εφαρμόζεται μια γάμμα κανονικοποίηση στην εικόνα ώστε να εξαλειφθούν έντονες περιοχές που δημιουργούνται λόγω του φωτισμού. Στη συνέχεια υπολογίζονται προσεγγιστικά οι παράγωγοι πρώτου βαθμού της εικόνας. Η εικόνα χωρίζεται σε κελιά συγκεκριμένου μεγέθους και για κάθε κελί δημιουργείται ένα ιστόγραμμα προσανατολισμού (orientation histogram). Αυτό το ιστόγραμμα χωρίζει το εύρος των γωνιών των παραγώγων σε ένα σταθερό αριθμό κάδων. Μέσω ενός συστήματος "ψηφοφορίας" γεμίζουν οι κάδοι και δημιουργείται το ιστόγραμμα. Η απόφαση για το σε τι ποσοστό θα συνεισφέρει κάθε παράγωγος σε κάθε κάδο γίνεται βάσει της γωνίας της παραγώγου, ενώ ο ακριβής αριθμός που θα συνεισφέρει καθορίζεται από το μέγεθος (magnitude) της παραγώγου. Τέλος, εφαρμόζεται μία ακόμη κανονικοποίηση σε επίπεδο μπλοκ (ομάδων από κελιά). Τα κανονικοποιημένα αυτά μπλοκ καλούνται HOG descriptors.

### 5.3.2 Υλοποίηση

Για την υλοποίηση του HOG δημιουργήθηκε η συνάρτηση **hog_fe**. Λειτουργεί παρόμοια με την **pca_fe** με την διαφορά ότι λαμβάνει ως είσοδο μόνο έναν dataloader καθώς η HOG για την εφαρμογή της δεν απαιτεί πληροφορία από άλλες εικόνες και κατά συνέπεια κανενός είδους εκπαίδευση. Για την εξαγωγή χαρακτηριστικών χρησιμοποιήθηκε η συνάρτηση **hog** της **skimage**.

### 5.3.3 Πειραματική αξιολόγηση

Παρατηρούμε ότι οι επιδόσεις της HOG είναι εξαιρετικά καλές (βλ. 11,10) αν και η ταχύτητα σύγκλισης είναι σχετικά αργή καθώς ο ταξινομητής συγκλίνει περίπου στις 30 εποχές (βλ. 9) για το 100% και το 50% του συνόλου εκπαίδευσης.

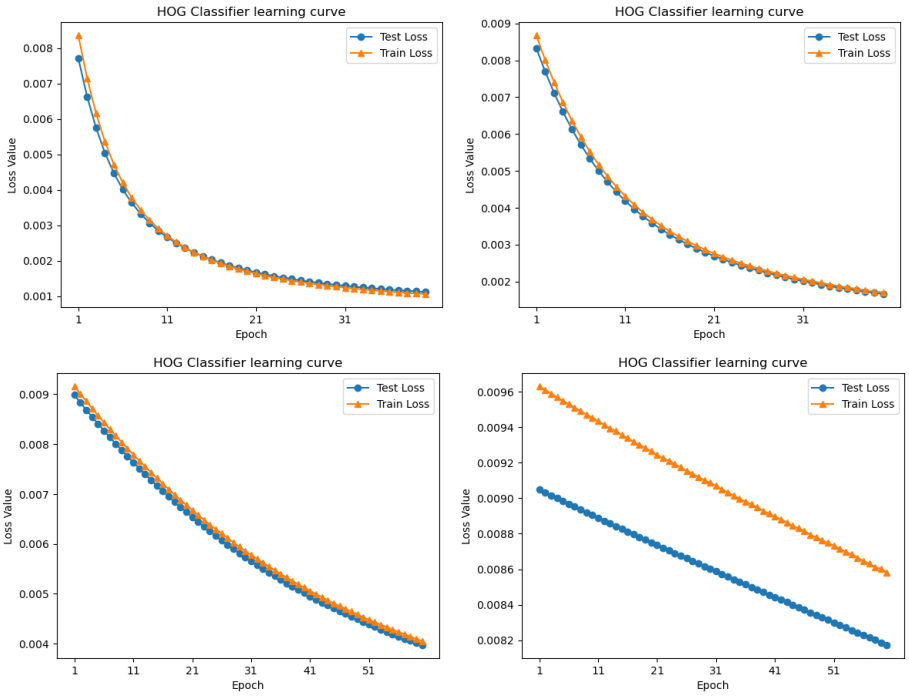
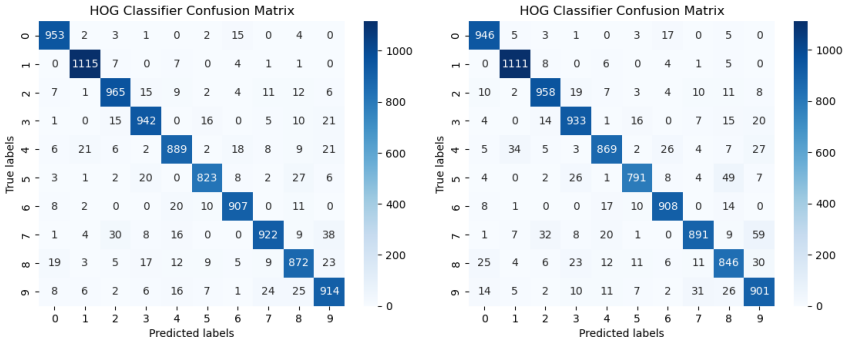



Figure 9: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Χρησιμοποιώντας το 10% του συνόλου εκπαίδευσης, η σύγκλιση γίνεται πολύ πιο αργά (60+ εποχές, βλ. 9). Από την άλλη οι επιδόσεις διατηρούνται αρκετά υψήλα (87%, βλ. 9). Μόνο χρησιμοποιώντας το 1% του συνόλου εκπαίδευσης παρατηρείται αποτυχία σύγκλισης και κατάρρευση των επιδόσεων. Ωστόσο, αυτή οφείλεται αμιγώς στην αδυναμία του ταξινομητή να εκπαιδευτεί με τόσα λίγα δείγματα, καθώς η HOG παράγει τα ίδια χαρακτηριστικά ανεξαρτήτως πλήθους δειγμάτων.



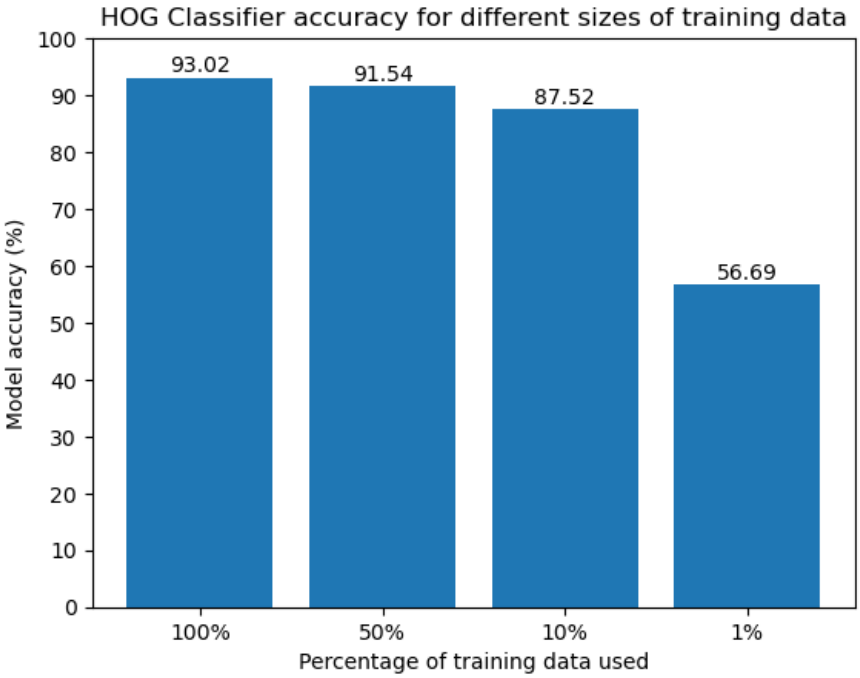



Figure 11: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

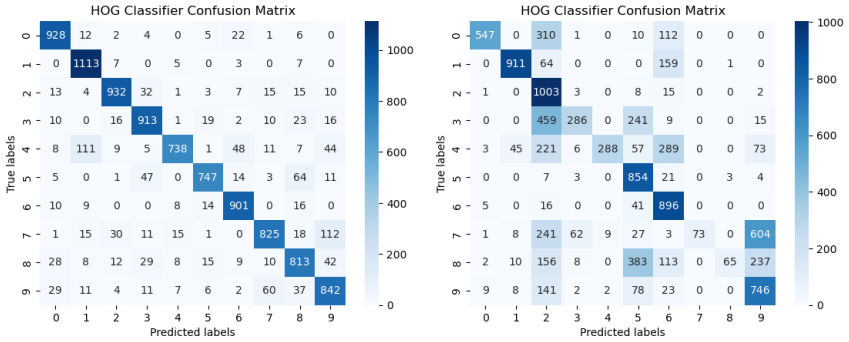



Figure 10: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

## 5.4 Binary Robust Independent Elementary Features (BRIEF)

### 5.4.1 Θεωρητικό υπόβαθρο

Τα BRIEF είναι δυαδικοί περιγραφείς χαρακτηριστικών. Πρόκειται για ταχύτερη και πιο συμπαγή μέθοδο από τη SIFT (που ανήκει στην ίδια κατηγορία μεθόδων) καθώς τα χαρακτηριστικά που παράγει είναι απλά μία δυαδική ακολουθία.

Για να παραχθεί αυτή η δυαδική ακολουθία απαιτείται πρώτα η επιλογή κάποιων σημείων στην εικόνα (keypoints). Στη συγκεκριμένη υλοποίηση επιλέξαμε να χρησιμοποιήσουμε τα σημεία των γωνιών στις εικόνες, χρησιμοποιώντας τον **Harris corner detector** για να τις εντοπίσουμε. Για κάθε keypoint η BRIEF υπολογίζει έναν περιγραφέα βάσει ενός μπλοκ εικονοστοιχείων γύρω από αυτό. Δειγματοληπτώντας ορισμένα ζεύγη πίξελ στο μπλοκ, υπολογίζει τον περιγραφέα ως εξής: Αν το πρώτο πίξελ του ζεύγους έχει μικρότερη τιμή έντασης από το δεύτερο, η αντίστοιχη τιμή στο διάνυσμα χαρακτηριστικών (περιγραφέας) γίνεται ίση με 1, αλλιώς γίνεται ίση με 0.

Ένα πρακτικό πρόβλημα που αντιμετωπίζει αυτή η μέθοδος είναι ότι αριθμός των keypoints που εντοπίζονται δεν είναι πάντα ο ίδιος. Αυτό σημαίνει πως το διάνυσμα χαρακτηριστικών που δημιουργείται δεν έχει σταθερό μήκος για όλα τα δείγματα και κατά συνέπεια δεν μπορεί να χρησιμοποιηθεί απευθείας από νευρωνικό ταξινομητή.

Για να αντιμετωπιστεί αυτό το πρόβλημα υπάρχουν μια σειρά λύσεις ώστε να μετατραπούν τα διανύσματα σε σταθερού μήκους: Bag of Visual Words (BoVW), Vector of Locally Aggregated Desctiptors (VLAD), Fisher Vectors (FVs) κ.α. . Εμείς επιλέξαμε την τελευταία λύση, η οποία θεωρείται ότι παράγει και τα καλύτερα αποτελέσματα.

Τα FVs χρησιμοποιούνται για τη δημιουργία διανυσμάτων χαρακτηριστικών σταθερού μήκους ως εξής:

1. Κατασκευάζεται ένα GMM [1] που μοντελοποιεί την κατανομή των δεδομένων.
2. Υπολογίζεται πώς οι παράμετροι της GMM (μέσοι όροι, διακυμάνσεις, βάρη) αλλάζουν, όταν παρατηρούνται τα δεδομένα.
3. Οι αλλαγές αυτές αποθηκεύονται σε ένα διάνυσμα (Fisher vector), που συνοψίζει την πληροφορία.

### 5.4.2 Υλοποίηση

Για την υλοποίηση της BRIEF, δημιουργήθηκε η συνάρτηση **brief_fe**, η οποία λειτουργεί με αντίστοιχο τρόπο με την **pca_fe**. Λαμβάνει ως είσοδο ένα dataloader

[1] Το Gaussian Mixture Model(GMM) είναι ένα soft clustering μοντέλο το οποίο θεωρεί ότι η πιθανότητα κάθε δείγματος να ανήκει σε μια κλάση ακολουθεί μια Gaussian κατανομη. Συνεπώς, δεν υπάρχει αυστηρή συσταδοποίηση αλλά πιθανοτική (π.χ. το δείγμα Χ είναι 80% Α και 20% Β)

και προαιρετικά ένα αντικείμενο GMM. Για κάθε εικόνα εντοπίζει τα keypoint που είναι οι κορφές των γωνιών που εντοπίζει ο Harris corner detector. Για να το πετύχει αυτό χρησιμοποιεί τις συναρτήσεις **corner_peaks** και **corner_harris** της skimage. Στη συνέχεια θα εξάγει του περιγραφείς των keypoints, με χρήση της κλάσης **BRIEF**, της **skimage**. Αν δε λάβει προεκπαιδευμένο GMM στην είσοδο (φάση εξαγωγής χαρακτηριστικών συνόλου εκπαίδευσης) θα δημιουργήσει και θα εκπαιδεύσει ένα αντικείμενο GMM πάνω στα δεδομένα εισόδου, χρησιμοποιώντας τη συνάρτηση **learn_gmm** της **skimage**. Στη συνέχεια χρησιμοποιώντας το εκπαιδευμένο GMM (ή αυτό που του δόθηκε ως είσοδος) θα εξάγει το fisher vector κάθε εικόνας χρησιμοποιώντας την **fisher_vector** της **skimage**.

### 5.4.3 Πειραματική αξιολόγηση

Η προφανής παρατήρηση που μπορεί να κάνει κάποιος κοιτώντας το διάγραμμα 14 είναι ότι οι επιδόσεις του μοντέλου δεν είναι ικανοποιητικές. Αυτό μας οδηγεί στο συμπέρασμα ότι τα χαρακτηριστικά που παράγονται από την BRIEF δεν είναι εύκολα διαχωρίσιμα με αποτέλεσμα ο ταξινομητής να μην μπορεί να τα ταξινομήσει σωστά. Αυτό το φαινόμενο θεωρούμε ότι οφείλεται στη συγκεκριμένη υλοποίηση και όχι στη μέθοδο καθεαυτή. Ενδεχομένως η επιλογή των keypoints να μην είναι κατάλληλη ή η επιλογή των παραμέτρων του GMM να μην είναι η βέλτιστη. Ίσως και η ίδια η αξιοποίηση των fisher vectors να μην ενδείκνυται για αυτού του είδους το πρόβλημα.

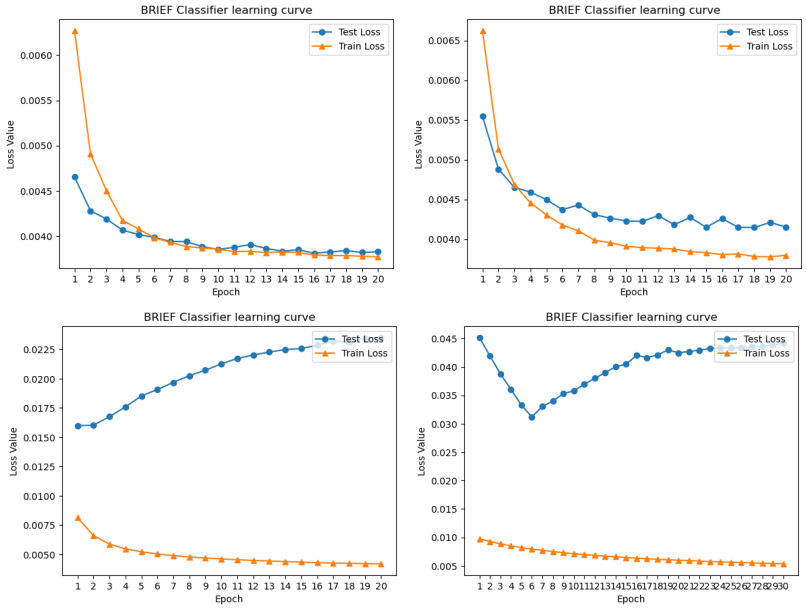



Figure 12: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Όσον αφορά τη σύγκλιση του μοντέλου, παρατηρούμε ότι η ταχύτητα σύγκλισης είναι συγκρίσιμη με τα υπόλοιπα μοντέλα για το 100% και το 50% του συνόλου εκπαίδευσης (περίπου 10 εποχές) αν και η σύγκλιση γίνεται σε πολύ μεγαλύτερη τιμή της συνάρτησης απώλειας, κάτι που αντανακλάται και στην ακρίβεια και στο μητρώο σύγχυσης (βλ. 13, 14). Ωστόσο, η μείωση του συνόλου εκπαίδευσης κατά 50% έχει πολύ μικρή επίπτωση στην ακρίβεια του μοντέλου.

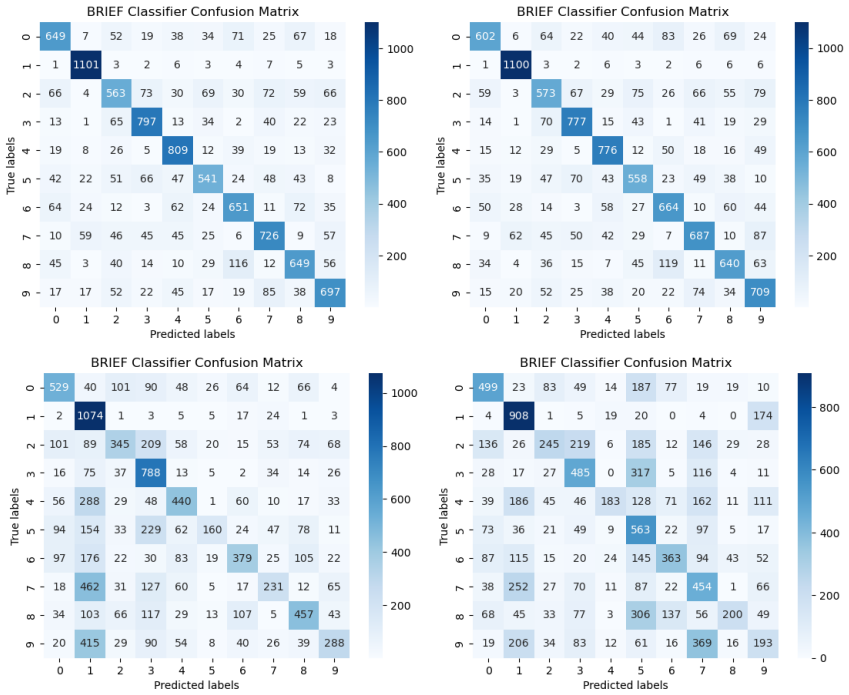



Figure 13: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Όσον αφορά τη χρήση του 10% και του 1% του συνόλου εκπαίδευσης, παρατηρείται αποτυχία σύγκλισης του μοντέλου (βλ. 12) με ανάλογη κατάρρευση της ακρίβειας και αποτύπωσης πολλαπλών λαθών στο μητρώο σύγχυσης (βλ. 14, 13).

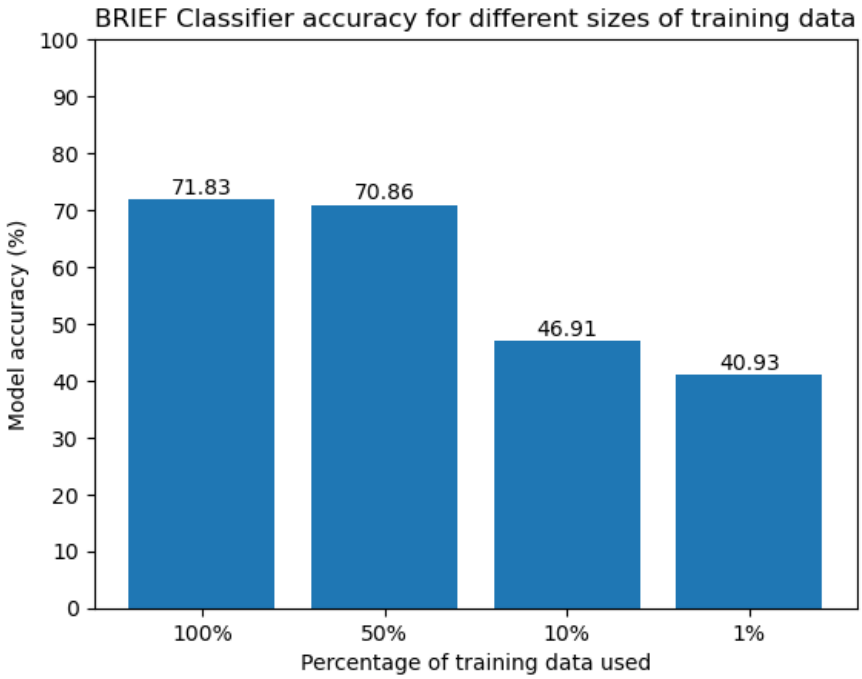



Figure 14: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

## 5.5 Autoencoder (AE)

### 5.5.1 Θεωρητικό υπόβαθρο

Οι αυτοκωδικοποιητές αποτελούν ένα είδος feed-forward νευρωνικού δικτύου που αποτελείται από δύο βασικά στοιχεία: έναν κωδικοποιητή και έναν αποκωδικοποιητή. Ο κωδικοποιητής έχει σχεδιαστεί για να συμπιέζει τα δεδομένα εισόδου σε μια αναπαράσταση χαμηλότερης διάστασης (bottleneck) ενώ ο ρόλος του αποκωδικοποιητή είναι να ανακατασκευάζει τα αρχικά δεδομένα από την έξοδο του κωδικοποιητή. Αυτή η διαδικασία συμπίεσης-αποσυμπίεσης πραγματοποιείται μέσω στρωμάτων με προοδευτικά μειούμενο αριθμό νευρώνων που στη συνέχεια αυξάνονται πίσω στην αρχική διάσταση στον αποκωδικοποιητή. Συνήθως, τα στρώματα αυτά έχουν συμμετρική δομή.

Στόχος είναι το νευρωνικό δίκτυο να διακρίνει τις πιο κρίσιμες διαστάσεις ή συσχετίσεις εντός των δεδομένων εισόδου. Εφόσον εκπαιδευτούν σωστά, οι αυτοκωδικοποιητές αντιλαμβάνονται τα ουσιώδη χαρακτηριστικά για την αναπαράσταση των δεδομένων και είναι χρήσιμοι σε εφαρμογές όπως η μείωση της διαστατικότητας και η εξαγωγή χαρακτηριστικών.

Μετά την εκπαίδευση του αυτοκωδικοποιητή, ο κωδικοποιητής του μπορεί να χρησιμοποιηθεί για την αναπαράσταση των δεδομένων σε μικρότερη διάσταση. Κατί τέτοιο, όπως και στην περίπτωση της PCA, ισοδυναμεί με εξαγωγή χαρακτηριστικών.

Οι γραμμικοί αυτοκωδικοποιητές έχουν γραμμικές συναρτήσεις ενεργοποίησης, πράγμα που ισοδυναμεί με απουσία συνάρτησης ενεργοποίησης καθώς η είσοδος περνά αυτούσια στην έξοδο. Αν η συνάρτηση απώλειας την οποία προσπαθεί να μηδενίσει ο γραμμικός αυτοκωδικοποιητής είναι το Μέσο Τετραγωνικό Σφάλμα, τότε καταλήγει εν τέλει να εντοπίζει τις κύριες συνιστώσες των δεδομένων εισόδου, δηλαδή να ισοδυναμεί με την PCA.

Για τους μη γραμμικούς κωδικοποιητές ισχύει ό,τι και για τους γραμμικούς με τη διαφορά ότι χρησιμοποιούν μη γραμμικές συναρτήσεις ενεργοποίησης. Ακριβώς αυτή τους η ιδιότητα τους επιτρέπει να εντοπίσουν μη γραμμικές συσχετίσεις μεταξύ των δεδομένων. Λόγω αυτού μπορούν να θεωρηθούν και γενίκευση της PCA.

### 5.5.2 Υλοποίηση

Για την υλοποίηση του autoencoder, δημιουργήθηκε η κλάση Autoencoder (κληρονομεί από τη nn.Module). Η δομή του αποτελείται από δύο συναρτήσεις (στιγμιότυπα της nn.Sequential), την encoder και την decoder.

Η **encoder** αποτελείται από ένα στρώμα μετατροπής της εισόδου (εικόνα) σε διάνυσμα (**nn.Flatten**) και από δύο πλήρως συνδεδεμένα στρώματα με συνάρτηση ενεργοποίησης τη ReLU, που μειώνουν τη διάσταση της εισόδου σε 128.

Η **decoder** εκτελεί την ανάποδη διαδικασία χρησιμοποιώντας αντίστοιχα στρώματα για να αυξήσει τη διάσταση της εισόδου στην αρχική της τιμή (28 επί 28) και ένα στρώμα για τον ανασχηματισμό της σε δύο διαστάσεις (**nn.Unflatten**).

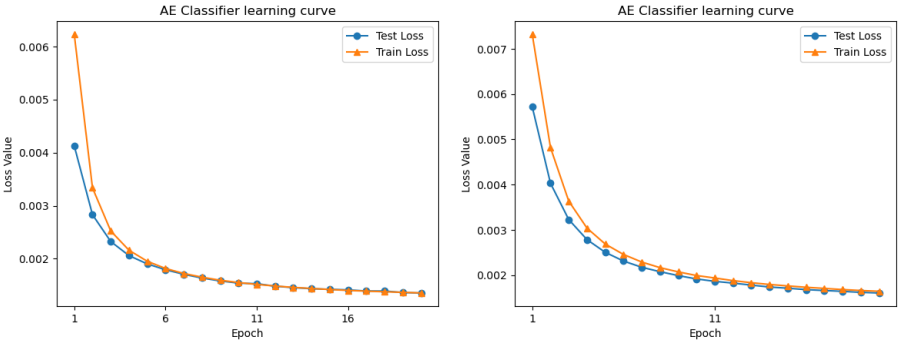
Η μέθοδος **forward**, περνάει την είσοδο πρώτα από τον encoder και μετά από τον decoder, επιστρέφοντας την έξοδο του τελευταίου.

Για την εκπαίδευση του autoencoder, υλοποιήθηκε η συνάρτηση **ae_train**, η οποία ακολουθεί παρόμοια λογική με την **nn_train** με τη διαφορά ότι στην είσοδό της, δίνουμε το μέσο τετραγωνικό σφάλμα (nn.MSELoss) ως συνάρτηση απώλειας και ότι αυτή υπολογίζεται όχι με βάση τα labels (αφού δεν έχουμε classification πρόβλημα) αλλά με βάση τη διαφορά εισόδου - εξόδου. Συνεπώς, εκπαιδεύουμε τον autoencoder ώστε να ανακατασκευάζει όσο το δυνατόν καλύτερα την εικόνα στην εξοδό του.

Για την εξαγωγή των χαρακτηριστικών δημιουργήθηκε η συνάρτηση **ae_fe**. Ακολουθεί αντίστοιχη λογική με **pca_fe**. Όταν δεν της δοθεί ένας εκπαιδευμένος autoencoder στην είσοδο, δημιουργεί και εκπαιδεύει έναν καινούργιο, τον οποίο στη συνέχεια αποθηκεύει ώστε να χρησιμοποιηθεί για την εξαγωγή των χαρακτηριστικών και των υποσυνόλων ελέγχου και επικύρωσης.

### 5.5.3 Πειραματική αξιολόγηση

Παρατηρούμε ότι με τα χαρακτηριστικά που εξάγονται από τον autoencoder, οι επιδόσεις του ταξινομητή είναι ευφάμηλες με αυτές της PCA για το 100% του συνόλου εκπαίδευσης (βλ. 17). Το ίδιο ισχύει και για ταχύτητα σύγκλισης (βλ. 15).



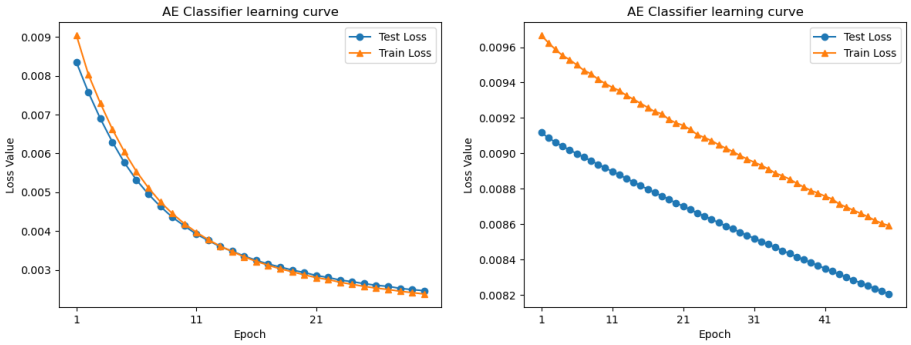



Figure 15: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Ωστόσο, όντας νευρωνικό δίκτυο είναι πολύ πιο επιρρεπές στη μείωση της ποσότητας των δεδομένων εκπαίδευσης. Η επιδόσεις του μειώνονται αισθητά όταν χρησιμοποιείται το 50% και το 10% του συνόλου εκπαίδευσης (βλ. 17), ενώ και ο χρόνος σύγκλισης σημειώνει την αναμενόμενη αύξηση(βλ. 15). Χρησιμοποιώντας το 1% του συνόλου εκπαίδευσης παρατηρούμε ότι το μοντέλου αποτυγχάνει να συγκλίνει και οι επιδόσεις του καταρρέουν όπως και στο CNN (βλ. 17).

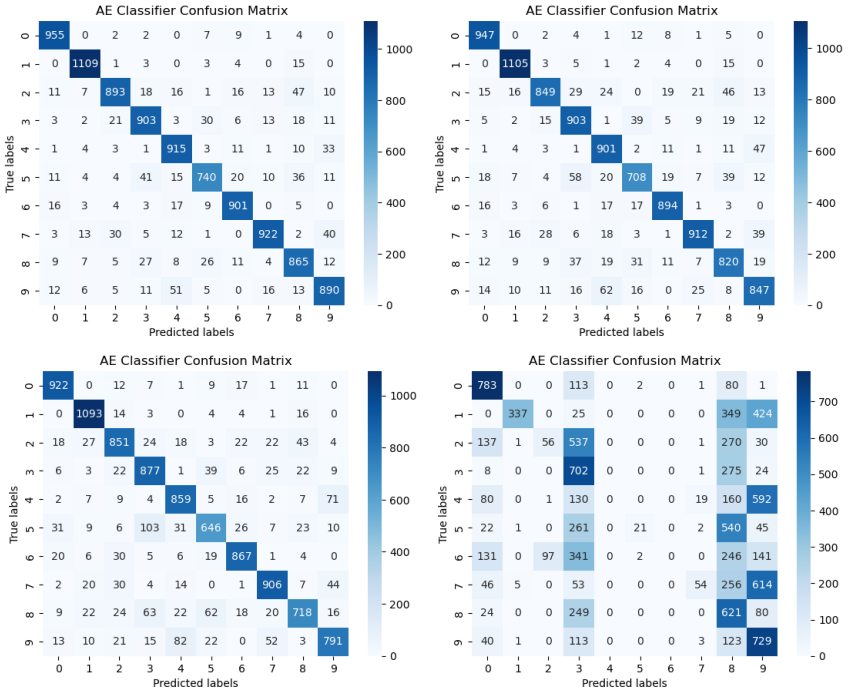


Figure 16: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

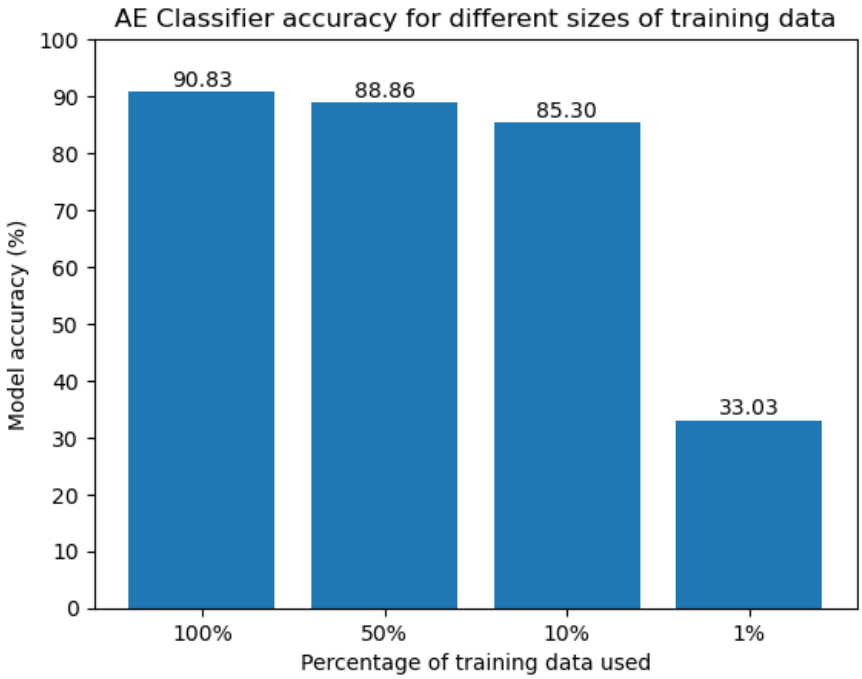



Figure 17: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

## 5.6 Vision Transformer (ViT)

### 5.6.1 Θεωρητικό υπόβαθρο

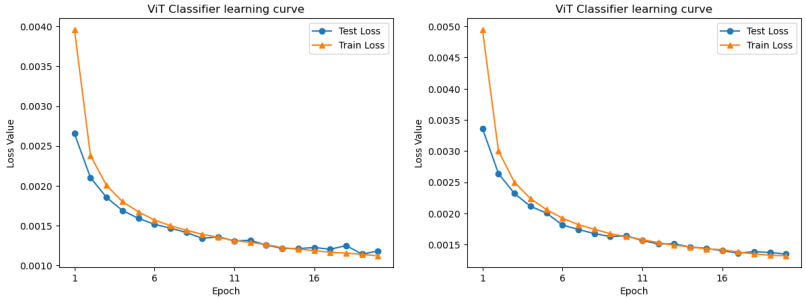
Μια κατηγορία νευρωνικών δικτύων που έχει αναδεχθεί τα τελευταία χρόνια και έχει συμβάλει στη ραγδαία ανάπτυξη της τεχνητής νοημοσύνης, είναι οι μετασχηματιστές (transformers). Αξιοποιώντας έναν μηχανισμό "προσοχής" (attention), οι μετασχηματιστές αντιλαμβάνονται το πλαίσιο (context) μέσα στο οποίο δίνεται μία είσοδος. Λόγω αυτής τους της ιδιότητας επιτυγχάνουν πολύ καλές επιδόσεις. Αν και αρχικά αξιοποιήθηκαν στα LLMs, τα τελευταία χρόνια έχουν αξιοποιηθεί και σε προβλήματα υπολογιστικής όρασης, με τη μορφή των Vision Transformers (ViT). Ένα από τα σημαντικότερα framework για την εκπαίδευση ViT είναι το DINOv2 της meta. Επιλέξαμε συνεπώς, να χρησιμοποιήσουμε ένα vits14 εκπαιδευμένο με DINOv2 ως εξαγωγέα χαρακτηριστικών.

### 5.6.2 Υλοποίηση

Η εξαγωγή χαρακτηριστικών γίνεται μέσω του script **ViT_FE.ipynb**, το οποίο προτείνεται να εκτελεστεί σε περιβάλλον google colab. Για να εξάγουμε τα χαρακτηριστικά φορτώνουμε το προεκπαιδευμένο μοντέλο από τις βιβλιοθήκες του pytorch και του φορτώνουμε τα δεδομένα του dataset μας. Δεν εκπαιδεύουμε το δίκτυο πάνω στα δικά μας δεδομένα καθώς, πρώτον αυτό θα απαιτούσε μεγάλη υπολογιστική ισχύ και θα ήταν μια χρονοβόρα διαδικασία και, δεύτερον το δίκτυο θεωρητικά μπορεί να χρησιμοποιηθεί για την εξαγωγή χαρακτηριστικών χωρίς περαιτέρω εκπαίδευσης (fine-tuning).

### 5.6.3 Πειραματική αξιολόγηση

Παρατηρούμε ότι αν και οι επιδόσεις του μοντέλου μπορούν να θεωρηθούν καλές, δεν ξεπερνούν αυτές του CNN (βλ. 20,5). Αυτό μπορεί αρχικά να φαντάζει παράξενο για ένα SoTA μοντέλο. Ωστόσο, πρέπει να λάβουμε υπόψη μας ότι το μοντέλο δεν έχει εκπαιδευτεί στα δεδομένα του προβλήματός μας, ενώ επιπλέον η μικρή διάσταση των δεδομένων (28 επί 28) μάλλον δεν ευνοεί την εξαγωγή χαρακτηριστικών από ένα τέτοιου είδους μοντέλο, το οποίο μάλιστα εξάγει και ένα σχετικά μεγάλο διάνυσμα χαρακτηριστικών (384 διαστάσεων).



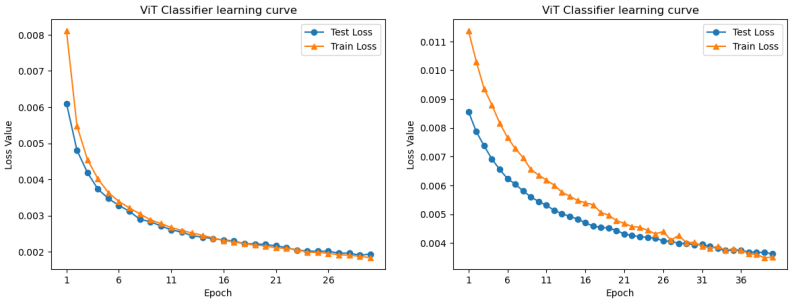



Figure 18: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Παρατηρούμε επιπλέον την αναμενόμενη διατήρηση της υψηλής ακρίβειας όταν χρησιμοποιείται το 50% και το 10% του συνόλου εκπαίδευσης (βλ. 20. Παρατηρούμε επίσης και την ανάλογη μείωση της ταχύτητας σύγκλισης (βλ. 18).

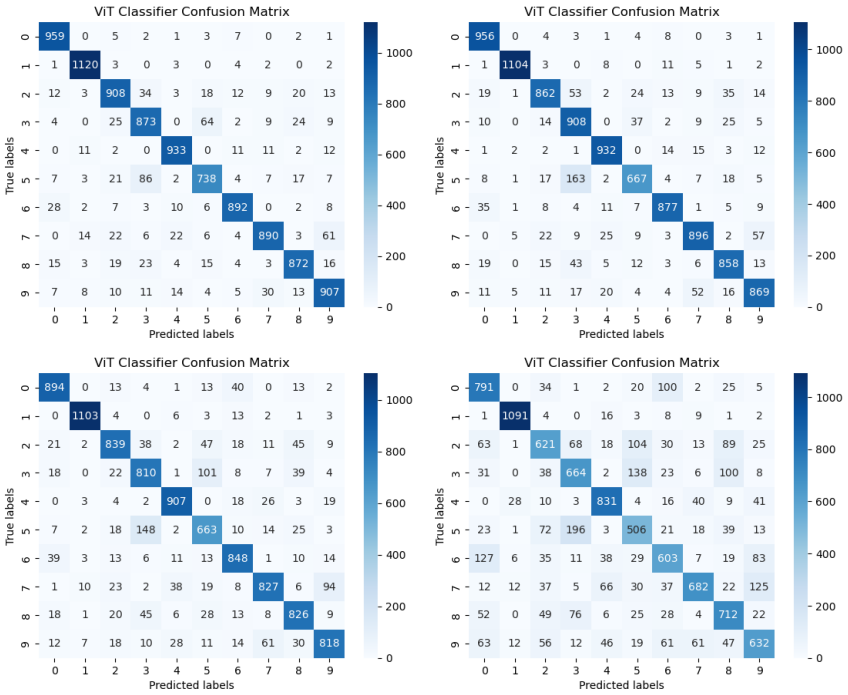



Figure 19: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

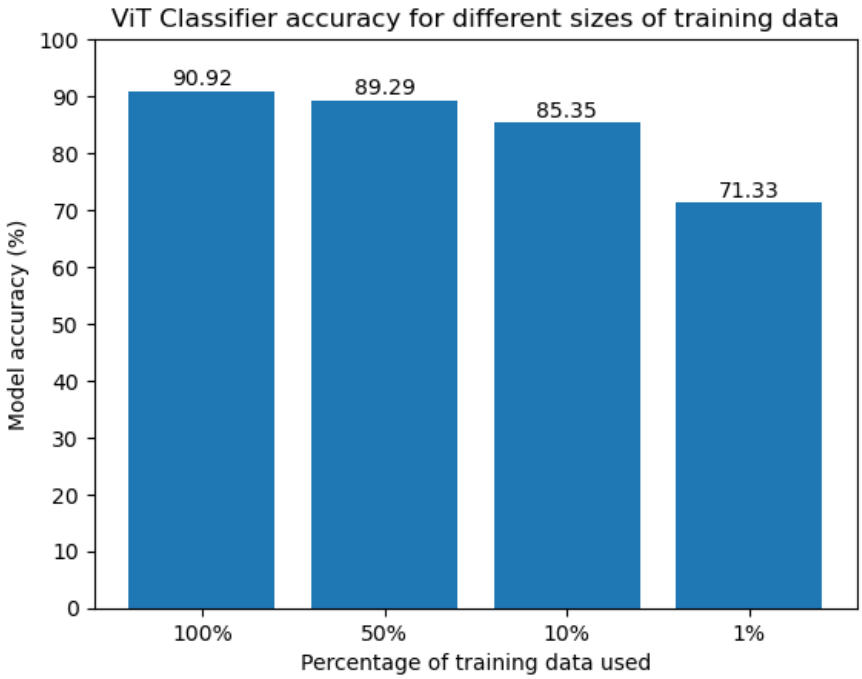



Figure 20: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

Ενδιαφέρον παρουσιάζει επίσης η περίπτωση που χρησιμοποιείται το 1% του συνόλου εκπαίδευσης. Εκεί παρατηρούμε ότι το μοντέλο και συγκλίνει και επιτυγχάνει ακρίβεια άνω του 50% (βλ. 18,20).

## 5.7 Fourier Descriptors (FD)

### 5.7.1 Θεωρητικό υπόβαθρο

Οι περιγραφείς Fourier χρησιμοποιούνται για την προσέγγιση του περιγράμματος ενός αντικειμένου. Πρόκειται για τον μετασχηματισμό Fourier των μιγαδικών αριθμών που προκύπτουν θέτοντας το μήκος του πραγματικού και του φανταστικού μέρους ίσο με τις συντεταγμένες των εικονοστοιχείων του περιγράμματος του αντικειμένου. Αφαιρώντας τις λιγότερο σημαντικές, δηλαδή τις υψηλότερες συχνότητες του μετασχηματισμού και αντιστρέφοντας την παραπάνω διαδικασία, μπορούμε να κατασκευάσουμε μια προσέγγιση του περιγράμματος αφαιρώντας τις περιττές λεπτομέρειες. Μια τέτοια προσέγγιση φαίνεται ταιριαστή στο πρόβλημά μας, καθώς αφορά εντοπισμό ψηφίων που αν και είναι γραμμένα διαφορετικά, ως προς το γενικό τους περίγραμμα θα πρέπει να μοιάζουν.

### 5.7.2 Υλοποίηση

Όσον αφορά την υλοποίηση, προέκυψαν δύο σημαντικά ζητήματα για την αξιοποίηση των περιγραφέων Fourier.

Πρώτον, ότι δεν έχουν όλες εικόνες το ίδιο μήκος περιγράμματος, οδηγώντας μας σε διαφορετικού μήκους διανύσματα χαρακτηριστικών. Αυτό διορθώθηκε με τη χρήση zero padding, η οποία αν και δεν είναι ιδανική, δε φαίνεται να επηρεάζει το τελικό αποτέλεσμα καθώς εφαρμόζεται σε πολύ λίγα δείγματα αφού χρησιμοποιείται μόλις ένα μέρος των συχνοτήτων του μετασχηματισμού Fourier, οπότε το μήκος του περιγράμματος σπάνια αποτελεί πρόβλημα.

Το δεύτερο ζήτημα που προέκυψε αφορούσε την τροφοδότηση των περιγραφέων στον ταξινομητή. Δεν μπορούμε να τροφοδοτήσουμε τον ταξινομητή με μιγαδικούς αριθμούς (όπως είναι οι συντελεστές του μετασχηματισμού Fourier), οπότε έπρεπε να βρούμε έναν τρόπο αναπαράστασής τους σε μη μιγαδική μορφή. Έπειτα από πειραματισμούς καταλήξαμε στην αναπαράστασή τους ως ένα διάνυσμα, στην αρχή του οποίου βρίσκονται τα πραγματικά μέρη και στο τέλος τα φανταστικά μέρη των συντελεστών. Αν και ίσως φαντάζει παράδοξο, αποδείχθηκε λειτουργικό.

Για το πρακτικό κομμάτι της υλοποίησης, δημιουργήθηκαν δύο συναρτήσεις η **compute_fourier_descriptors** και η **fd_fe**.

Η **compute_fourier_descriptors** λαμβάνει ως είσοδο μια εικόνα και αφού υπολογίσει το ιδανικό κατώφλι με τη μέθοδο Otsu με χρήση της συνάρτησης **threshold_otsu**, κατωφλιώνει την εικόνα ώστε να μετατρέψει το σχήμα σε απόλυτο άσπρο και το υπόβαθρο σε απόλυτο μαύρο. Στη συνέχεια χρησιμοποιώντας την

**findContours** της **cv2** εντοπίζει το περίγραμμα του σχήματος. Αφού εντοπίσει το περίγραμμα, δημιουργεί τις μιγαδικές συντεταγμένες και εφαρμόζει πάνω τους τον μετασχηματισμό Fourier χρησιμοποιώντας την **fft** της **scipy**. Από τους περιγραφείς Fourier που προκύπτουν, διατηρεί τους 100 πρώτους (τις χαμηλότερες συχνότητες) και σε περίπτωση που είναι λιγότεροι, εφαρμόζει zero padding. Τέλος αναπαριστά τους περιγραφείς όπως αναφέρθηκε παραπάνω και τους επιστρέφει.

Η **fd_fe** λειτουργεί με αντίστοιχο τρόπο με την **hog_fe**, καλώντας την **compute_fourier_descriptors** για να εξάγει τα χαρακτηριστικά.

### 5.7.3 Πειραματική αξιολόγηση

Τόσο από τα μητρώα σύγχησης όσο και από την ακρίβεια του μοντέλου (βλ. 22, 23) φαίνεται ότι τα χαρακτηριστικά που εξάγουμε με τους περιγραφείς Fourier είναι κατάλληλα για αυτό το πρόβλημα ταξινόμησης.

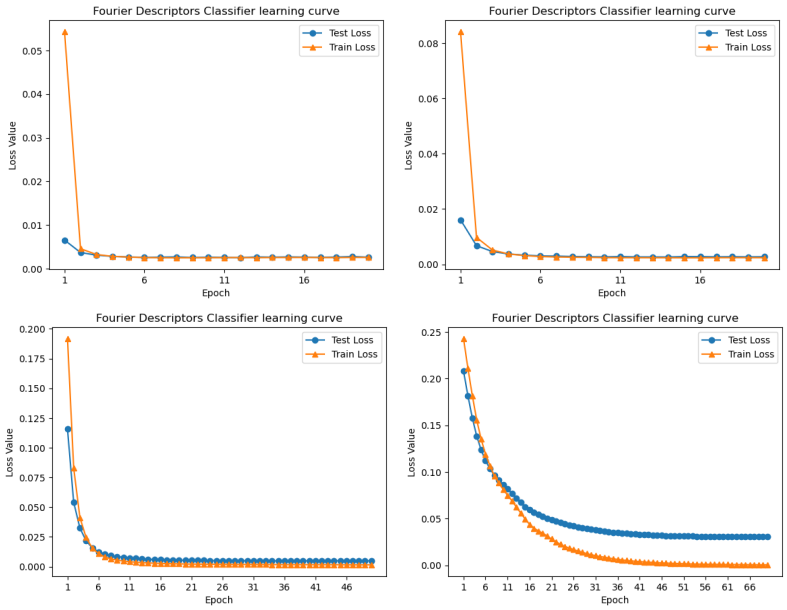



Figure 21: Καμπύλη μάθησης του δικτύου για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Η ταχύτητα σύγκλισης είναι εντυπωσιακή. Για το 100% και το 50% του συνόλου εκπαίδευσης φαίνεται να επιτυγχάνεται σύγκλιση σε 3-4 εποχές, ενώ για το 10% επιτυγχάνεται περίπου στις 10 εποχές. Για το 1% του συνόλου, η σύγκλιση επιτυγχάνεται στις 30 εποχές.

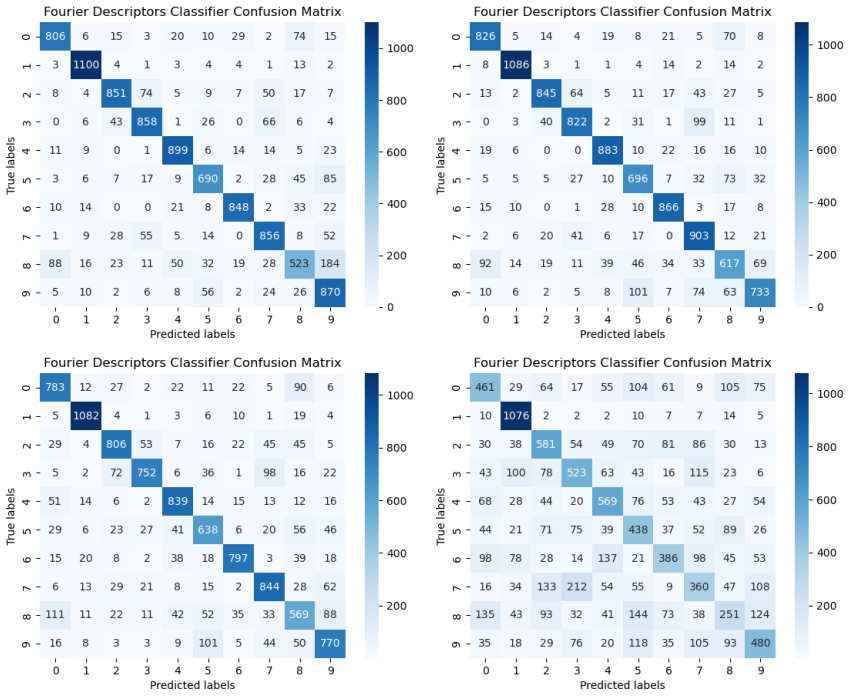



Figure 22: Μητρώο σύγχυσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης. Από αριστέρα προς τα δεξιά και από πάνω προς τα κάτω: 100%, 50%, 10% και 1% του αρχικού συνόλου.

Όσον αφορά την ακρίβεια του μοντέλου, αυτή δεν είναι ιδιαίτερα υψηλή αλλά ακολουθεί το μοτίβο που ακολουθείται στις περισσότερες περιπτώσεις: Η ακρίβεια μένει σχετικά σταθερή μέχρι να χρησιμοποιηθεί μόνο το 1% του συνόλου εκπαίδευσης. Εδώ η χαμηλή ακρίβεια οφείλεται αποκλειστικά στον ταξινομητή καθώς οι περιγραφείς Fourier δε χρειάζονται κάποιου είδους εκπαίδευσης για να εξαχθούν.

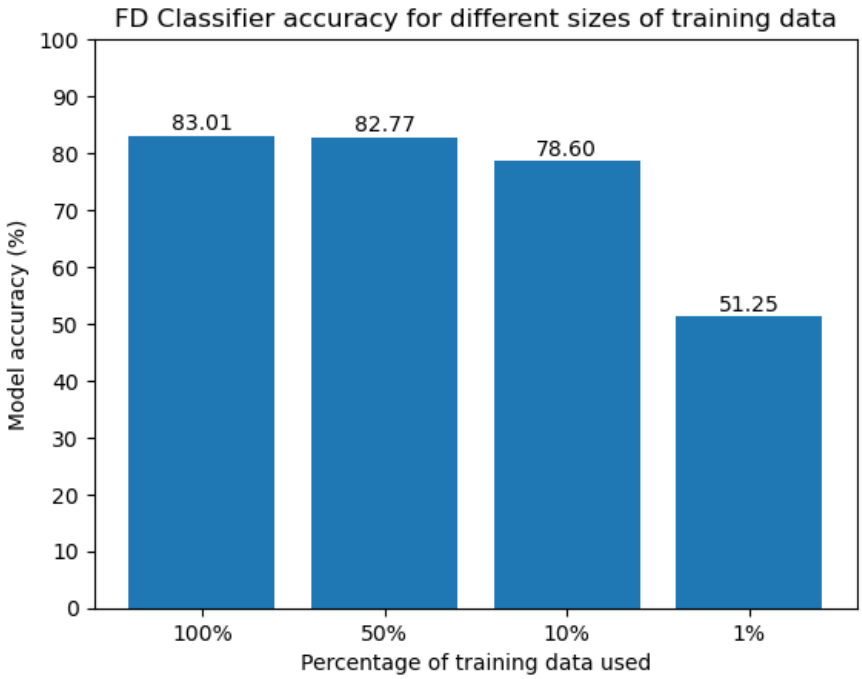



Figure 23: Η ακρίβεια του εκπαιδευμένου μοντέλου στο σύνολο επαλήθευσης για διαφορετικά μεγέθη συνόλου εκπαίδευσης

## 5.8 Συγκριτική αξιολόγηση

Η πρώτη παρατήρηση που οφείλουμε να κάνουμε αφορά την κυριαρχία του Συνελικτικού Νευρωνικού Δικτύου. Οι επιδόσεις του είναι σαφώς ανώτερες από όλες τις άλλες μεθόδους τόσο για το 100% όσο και το 50% του συνόλου εκπαίδευσης. Ακόμα και χρησιμοποιώντας το 10% του συνόλου εκπαίδευσης, καταφέρνει να επιτύχει υψηλή ακρίβεια σχεδόν ίση με την PCA που βρίσκεται στην πρωτιά σε αυτή την περίπτωση. Αυτές οι καλές επιδόσεις οφείλονται στην προσαρμοστικότητα της μεθόδου. Αναλόγως το πρόβλημα που έχουμε να αντιμετωπίσουμε (εδώ ταξινόμηση) και αναλόγως τα δεδομένα μας, το νευρωνικό δίκτυο θα προσαρμόσει κατάλληλα τα βάρη του ώστε να βελτιστοποιήσει τις επιδόσεις του στο συγκεκριμένο πρόβλημα. Αν ωστόσο προσπαθούσαμε να αξιοποιήσουμε το ίδιο δίκτυο για ένα άλλο πρόβλημα ταξινόμησης (π.χ. διαχωρισμός σκύλου / γάτας), θα παρατηρούσαμε την αδυναμία του να ανταπεξέλθει καθώς δεν έχει εκπαιδευτεί σε αντίστοιχα δεδομένα. Επιπλέον, το αρνητικό αυτής της μεθόδου είναι ότι είναι υπολογιστικά ακριβότερη σε σχέση με τις υπόλοιπες καθώς απαιτεί την εκπαίδευση ολόκληρου του δικτύου και όχι απλώς ενός νευρωνικού ταξινομητή.

Μια δεύτερη σημαντική παρατήρηση αφορά την ανθεκτικότητα της PCA στη μείωση των δεδομένων εκπαίδευσης. Παρατηρούμε ότι είναι η μόνη μέθοδος που διατηρεί περίπου 80% ακρίβεια ακόμη και χρησιμοποιώντας το 1% του συνόλου εκπαίδευσης. Ένα ακόμη συμπέρασμα που προκύπτει από αυτή τη διαδικασία είναι η συνάφεια συμπίεσης και εξαγωγής χαρακτηριστικών. Αυτό που κάνει η PCA είναι να προβάλει τα δεδομένα σε ένα μικρότερο χώρο, αυτό των κυρίων συνιστωσών. Κάτι αντίστοιχο πραγματοποιεί και ο autoencoder. Αντίστοιχα, η εξαγωγή χαρακτηριστικών οδηγεί και αυτή στην απεικόνιση των εικόνων σε χώρο μικρότερης διάστασης. Συνεπώς, οι δύο διαδικασίες οδηγούν σε παρόμοια αποτελέσματα και γι' αυτό μπορούν να χρησιμοποιηθούν στην ταξινόμηση.

Οφείλουμε επίσης να παρατηρήσουμε την εντυπωσιακή επίδοση της HOG, η οποία επιτυγχάνει ακρίβεια πολύ κοντά σε αυτή του ΣΝΔ, χωρίς μάλιστα να απαιτεί κανενός είδους εκπαίδευση ή πρότερη "γνώση" των δεδομένων, πέραν της εκπαίδευσης του ταξινομητή.

Τέλος, μια γενικότερη παρατήρηση που οφείλουμε να κάνουμε αφορά τη διάκριση μεταξύ των δύο κατηγοριών των μεθόδων που επιλέξαμε. Η πρώτη αφορά τις μεθόδους που δεν απαιτούν καμία πρότερη "γνώση" των δεδομένων για να περιγράψουν / εξάγουν τα χαρακτηριστικά (HOG, FD, ViT[2]) και η δεύτερη σε αυτές που απαιτούν κάποιου είδους εκπαίδευση πριν την εξαγωγή χαρακτηριστικών (CNN, BRIEF[3], PCA, AE). Παρατηρούμε ότι όταν μειώνεται κατά πολύ το σύνολο εκπαίδευσης (1% του αρχικού), αν και όλες οι μέθοδοι εμφανίζουν μειωμένες επιδόσεις, στις μεθόδους τις δεύτερης κατηγορίας παρατηρούνται ραγδαίες

[2] Αν και δεν απαιτεί πρότερη γνώση των δικών μας δεδομένων, έχει προεκπαιδευτεί σε άλλα σύνολα δεδομένων για να μπορεί να εξάγει χαρακτηριστικά.

[3] Αν και BRIEF καθεαυτή δεν απαιτεί εκπαίδευση, την απαιτεί η χρήση του GMM γι' αυτό και τη συγκαταλέγουμε σε αυτή την κατηγορία

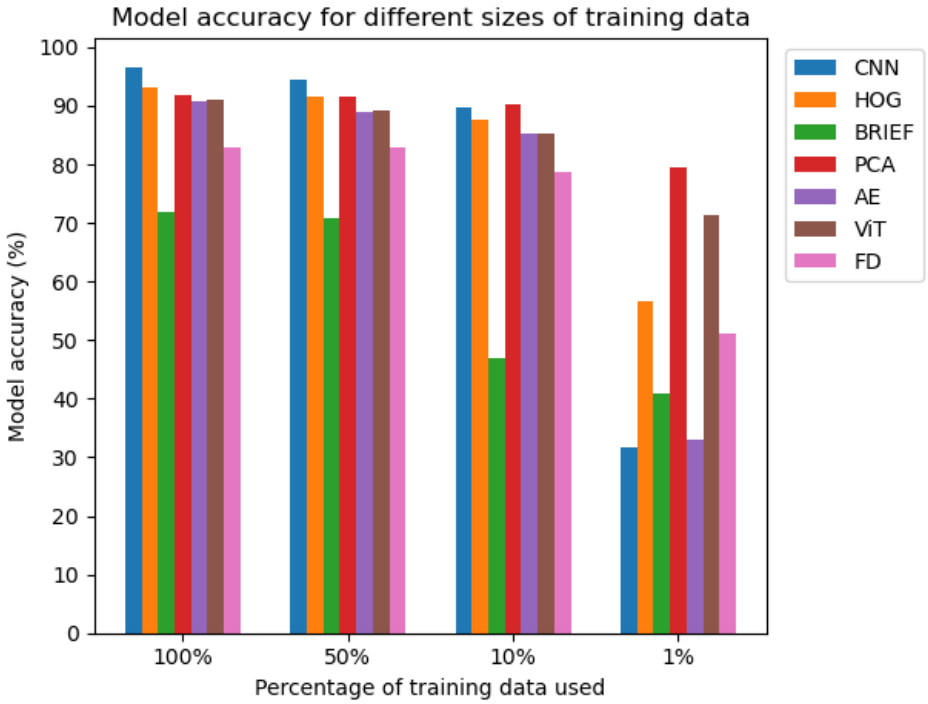



Figure 24: Σύγκριση της ακρίβειας των μοντέλων για διαφορετικά μεγέθη συνόλου εκπαίδευσης.

μειώσεις στην ακρίβεια. Αυτό οφείλεται στο ότι πέραν της αδυναμίας σωστής εκπαίδευσης του ταξινομητή (πράγμα που αφορά και την πρώτη κατηγορία), η εκπαίδευση των ίδιων των μεθόδων δεν δύναται να γίνει σωστά ώστε να συγκλίνουν και να μπορούν να γενικεύσουν. Εξαίρεση σε αυτό το μοτίβο αποτελεί η PCA, η οποία τόσο λόγω της φύσης της ως μέθοδος όσο και λόγω των δεδομένων (δεν παρουσιάζουν πολύ μεγάλες διαφορές), απαιτεί πολύ λιγότερα δεδομένα για να εντοπίσει τις κύριες συνιαστώσεις που απαιτεί για να μειώσει τη διαστατικότητα.

# 6 Βιβλιογραφία

# References

[1] R. C. Gonzalez, "Deep convolutional neural networks [lecture notes]", *IEEE Signal Processing Magazine*, vol. 35, no. 6, pp. 79–87, 2018.

[2] S. Haykin, *Νευρωνικά Δίκτυα και Μηχανική Μάθηση*. Παπασωτηρίου, 2009, pp. 373–380.

[3] D. Charte, F. Charte, S. García, M. J. del Jesus, and F. Herrera, "A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines", *Information Fusion*, vol. 44, pp. 78–96, 2018, ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.12.007. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253517307844.

[4] A. F. Agarap, *Implementing an autoencoder in pytorch*, Apr. 2023. [Online]. Available: https://medium.com/pytorch/implementing-an-autoencoder-in-pytorch-19baa22647d1.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Ieee, vol. 1, 2005, pp. 886–893.

[6] D. Tyagi, *Introduction to brief(binary robust independent elementary features)*, Apr. 2020. [Online]. Available: https://medium.com/@deepanshut041/introduction-to-brief-binary-robust-independent-elementary-features-436f4a31a0e6.

[7] Apr. 2024. [Online]. Available: https://en.wikipedia.org/wiki/Fisher_kernel.

[8] Sep. 2024. [Online]. Available: https://en.wikipedia.org/wiki/Vision_transformer.

[9] [Online]. Available: https://dinov2.metademolab.com/.

# Παράρτημα: Δημιουργία περιβάλλοντος και εκτέλεση κώδικα

Για την δημιουργία του περιβάλλοντος εκτέλεσης προτείνεται η αξιοποίηση του conda. Για να δημιουργηθεί το κατάλληλο περιβάλλον conda πρέπει να εκτελεστούν οι παρακάτω γραμμές:

```
conda env create -f env.yml
conda activate cnn
```

Για την εκτέλεση του κώδικα πρέπει να εκτελεστεί το αρχείο main.py με τα ανάλογα flags:

```
python main.py
--model ['cnn','pca','hog','brief','ae','vit','fd']
--epochs [int]
--data_pecentage [1,0.5,0.1,0.01]
--train
```

Απουσία του flag –train σημαίνει απλή αξιολόγηση του μοντέλου που βρίσκεται στο φάκελο *"output/model.pt.*