

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΒΑΘΙΑ ΜΑΘΗΣΗ

Ενδιάμεση Εργασία

Κουκουλέτσου Αικατερίνη 10218

Νοέμβριος 2024

Εισαγωγή

Η παρούσα εργασία έχει ως στόχο την σύγκριση της απόδοσης δύο μοντέλων ταξινόμησης, του k-Nearest Neighbors (KNN) με $k = 1$ και με $k = 3$ και του Nearest Centroid. Στόχος είναι να αξιολογηθεί η ακρίβεια ταξινόμησης των δύο μεθόδων, διερευνώντας την επίδραση διαφορετικών τιμών του k στο KNN και συγκρίνοντάς τις με την προσέγγιση των κεντροειδών.

Στο πρώτο μέρος της εργασίας, υλοποιούνται οι δύο ζητούμενοι αλγόριθμοι ταξινόμησης ενώ στο δεύτερο μέρος ακολουθεί μια εκτενέστερη μελέτη και υλοποίηση μεθόδων και τεχνικών που έχουν ως στόχο την βελτίωση της απόδοσης της ταξινόμησης.

Επιλογή Δεδομένων και Προεπεξεργασία

Το σύνολο δεδομένων που επιλέχθηκε για την ανάλυση είναι το CIFAR-10. Το CIFAR-10 περιλαμβάνει 60.000 έγχρωμες εικόνες διαστάσεων 32×32 , κατανεμημένες σε 10 κατηγορίες, με 6.000 εικόνες ανά κατηγορία. Κάθε εικόνα περιέχει τρία κανάλια χρώματος (red, green, blue), το οποίο συνεπάγεται ότι το αρχικό σχήμα των δεδομένων διαμορφώνεται ως εξής:

1. Για το **training set**: (num_samples, height, width, channels) \rightarrow (50.000, 32, 32, 3)
2. Για το **test set**: (num_samples, height, width, channels) \rightarrow (10.000, 32, 32, 3)

Αναφέρεται για λόγους πληρότητας ότι οι 10 κλάσεις του συνόλου δεδομένων CIFAR-10 είναι οι εξής: 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship' και 'truck'.

Προεπεξεργασία των Δεδομένων

Για την προεπεξεργασία των δεδομένων εφαρμόστηκε κανονικοποίηση και reshaping. Δεν χρειάστηκε να γίνει διαχωρισμός του dataset, καθώς το CIFAR-10 παρέχει έτοιμα τα σύνολα εκπαίδευσης και ελέγχου όπως αναφέρθηκε προηγουμένως.

Κανονικοποίηση

Η κανονικοποίηση των δεδομένων εξασφαλίζει ότι όλα τα χαρακτηριστικά θα βρίσκονται σε παρόμοια κλίμακα. Διαφορετικά δίχως κανονικοποίηση, χαρακτηριστικά με μεγαλύτερες τιμές (π.χ. 255 από τα κανάλια χρώματος) ενδέχεται να έχουν δυσανάλογη επίδραση στους υπολογισμούς των αποστάσεων, ανεξάρτητα από την σημασία τους για το μοντέλο. Αυτό μπορεί να προκαλέσει την «προτίμηση» του αλγόριθμου προς ορισμένες διαστάσεις των δεδομένων και να μειώσει την απόδοση της ταξινόμησης.

Ειδικά για αλγόριθμους όπως αυτός του kNN που βασίζεται στον υπολογισμό αποστάσεων των σημείων, έλλειψη κανονικοποίησης μπορεί να επηρεάσει δυσανάλογα τους υπολογισμούς, οδηγώντας σε κακή απόδοση, καθώς η απόσταση μπορεί να μην αντικατοπτρίζει τη πραγματική ομοιότητα των εικόνων.

Αντίστοιχα για τον Nearest Centroid, η κανονικοποίηση εξίσου συμβάλλει στην ακρίβεια, διότι ο αλγόριθμος υπολογίζει τον κεντροειδή κάθε κατηγορίας με βάση τις μέσες τιμές των χαρακτηριστικών. Χωρίς κανονικοποίηση, χαρακτηριστικά με υψηλότερες τιμές θα

επηρεάζαν υπερβολικά τη θέση του κεντροειδούς, προκαλώντας στρεβλώσεις στην ταξινόμηση. Η κανονικοποίηση γίνεται με διαίρεση με το 255 γιατί το 255 είναι η μέγιστη τιμή ενός pixel σε μια εικόνα 8-bit.

Reshaping - Flattening

Η διαδικασία "flattening" περιλαμβάνει τη μετατροπή πολυδιάστατων δεδομένων σε μονοδιάστατο διάνυσμα. Η τεχνική αυτή είναι απαραίτητη, καθώς τόσο το k-NN όσο και το Nearest Centroid λειτουργεί με διανύσματα χαρακτηριστικών, απαιτώντας κάθε δεδομένο να αναπαρίσταται ως μονοδιάστατο διάνυσμα στον υψηλής διάστασης χώρο. Για τον λόγο αυτό, γίνεται reshaping των δεδομένων από $(n, 32, 32, 3)$ σε $(n, 32 \cdot 32 \cdot 3)$ ή $(n, 3072)$.

Για το reshape των δεδομένων χρησιμοποιήθηκε η συνάρτηση `reshape((x_train.shape[0], -1))`. Το `x_train.shape[0]` αναφέρεται στον αριθμό των δειγμάτων εικόνων στο σύνολο εκπαίδευσης, ενώ η τιμή `-1` λειτουργεί ως placeholder, επιτρέποντας στο NumPy να καθορίσει αυτόματα το μέγεθος της δεύτερης διάστασης με βάση το συνολικό μέγεθος του array. Με αυτόν τον τρόπο, κάθε εικόνα μετατρέπεται σε μονοδιάστατο διάνυσμα, διατηρώντας παράλληλα το συνολικό πλήθος δεδομένων.

Μετρικές Αξιολόγησης

Στις παρακάτω υλοποιήσεις του αλγορίθμου χρησιμοποιούνται κάθε φορά οι ακόλουθες μετρικές αξιολόγησης.

1. Confusion Matrix

Confusion Matrix[i, j] = ο αριθμός των δειγμάτων της κατηγορίας i που ταξινομήθηκαν στην κατηγορία j

1. **Διαγώνια στοιχεία:** Τα διαγώνια στοιχεία του πίνακα δείχνουν τον αριθμό των σωστών προβλέψεων του μοντέλου για κάθε κατηγορία. Όσο μεγαλύτερη είναι η τιμή σε αυτά τα σημεία, τόσο καλύτερη είναι η απόδοση του μοντέλου. Συγκεκριμένα και σύμφωνα με την εικόνα, με τον όρο True Positive (TP) εννοείται ο αριθμός των θετικών παραδειγμάτων που ταξινομήθηκαν σωστά ως θετικά και με τον όρο True Negative (TN), ο αριθμός των αρνητικών παραδειγμάτων που ταξινομήθηκαν σωστά ως αρνητικά.
2. **Off-diagonal elements:** Οι τιμές που δεν βρίσκονται στη διαγώνιο δείχνουν λανθασμένες προβλέψεις. Όσο μεγαλύτερες είναι αυτές οι τιμές, τόσο πιο συχνά το μοντέλο κάνει λανθασμένες προβλέψεις και συγχέει τις κατηγορίες μεταξύ τους. Συγκεκριμένα και σύμφωνα με την εικόνα, με τον όρο False Positive (FP) εννοείται ο αριθμός των αρνητικών παραδειγμάτων που ταξινομήθηκαν λανθασμένα ως θετικά και με τον όρο False Negative (FN) Ο αριθμός των θετικών παραδειγμάτων που ταξινομήθηκαν λανθασμένα ως αρνητικά.

		Predicted Condition	
		Positive(PP)	Negative(PN)
Actual Condition	Total Population = P + N	True Positive(TP)	False Negative(FN)
	Positive(P)	False Positive(FP)	True Negative(TN)
	Negative(N)		

2. Classification Report

Το *Classification Report* παρέχει σημαντικές μετρικές αξιολόγησης για κάθε κατηγορία, όπως το Precision, Recall και F1-Score. Οι τύποι για αυτές τις μετρικές είναι οι εξής:

- **Precision:** Η αναλογία των σωστών θετικών προβλέψεων προς το σύνολο των προβλέψεων για την κατηγορία αυτή:

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

- **Recall:** Η αναλογία των σωστών θετικών προβλέψεων προς το σύνολο των πραγματικών θετικών παραδειγμάτων:

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

- **F1-Score:** Ο αρμονικός μέσος όρος της ακρίβειας και της ανακάλυψης:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3. Accuracy

Υπολογίζεται ως η αναλογία των σωστών προβλέψεων προς τον συνολικό αριθμό των παραδειγμάτων:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Υλοποίηση K-Nearest Neighbors Classifier με Scikit-Learn

1. **Ορισμός του k** Επιλέγεται ένας αριθμός γειτόνων, k , ο οποίος καθορίζει πόσοι γείτονες θα ληφθούν υπόψη για την κατηγοριοποίηση του νέου δείγματος.
2. **Υπολογισμός Απόστασης** Για το νέο δείγμα, υπολογίζεται η απόσταση του από όλα τα δείγματα του συνόλου εκπαίδευσης.
3. **Επιλογή k Κοντινότερων Γειτόνων** Οι αποστάσεις αυτές ταξινομούνται κατά αύξουσα σειρά και επιλέγονται οι k μικρότερες.
4. **Κατηγοριοποίηση:** Η κατηγορία του νέου δείγματος θα είναι τελικά η κατηγορία που υποδεικνύεται τις περισσότερες φορές από τα k δείγματα που είχαν τις μικρότερες αποστάσεις από το δείγμα x που ταξινομείται.

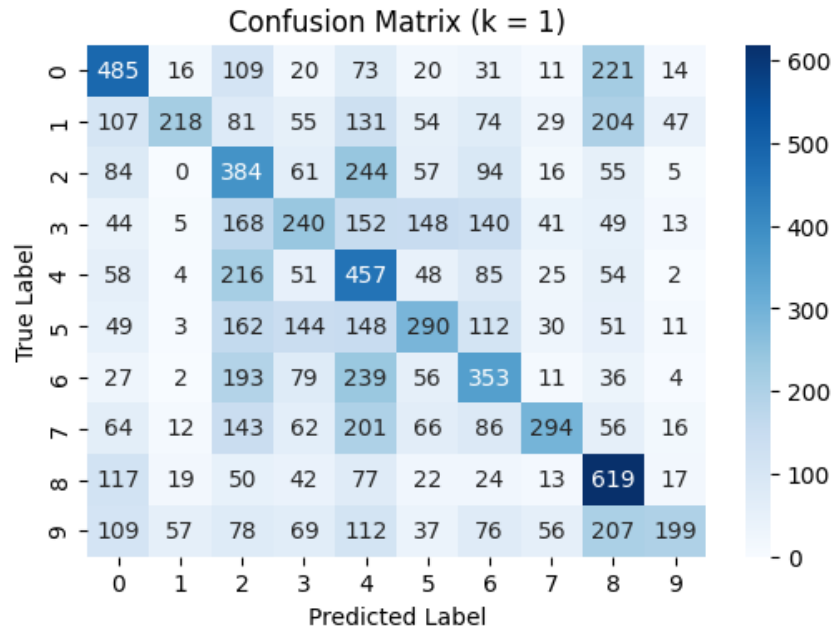
Σημειώνεται ότι, συνήθως, το k επιλέγεται ως περιττός αριθμός, ώστε να αποφεύγονται περιπτώσεις ισοβαθμίας κατά την κατηγοριοποίηση και ότι για τον υπολογισμό της απόστασης, χρησιμοποιείται συνήθως η Ευκλείδεια απόσταση.

k ίσο με 1

Accuracy Score 0.3539

Class	Precision	Recall	F1-Score	Support
Airplane	0.42	0.48	0.45	1000
Automobile	0.65	0.22	0.33	1000
Bird	0.24	0.38	0.30	1000
Cat	0.29	0.24	0.26	1000
Deer	0.25	0.46	0.32	1000
Dog	0.36	0.29	0.32	1000
Frog	0.33	0.35	0.34	1000
Horse	0.56	0.29	0.39	1000
Ship	0.40	0.62	0.49	1000
Truck	0.61	0.20	0.30	1000
Accuracy		0.35		10000
Macro Avg	0.41	0.35	0.35	10000
Weighted Avg	0.41	0.35	0.35	10000

Table 1: Classification Report

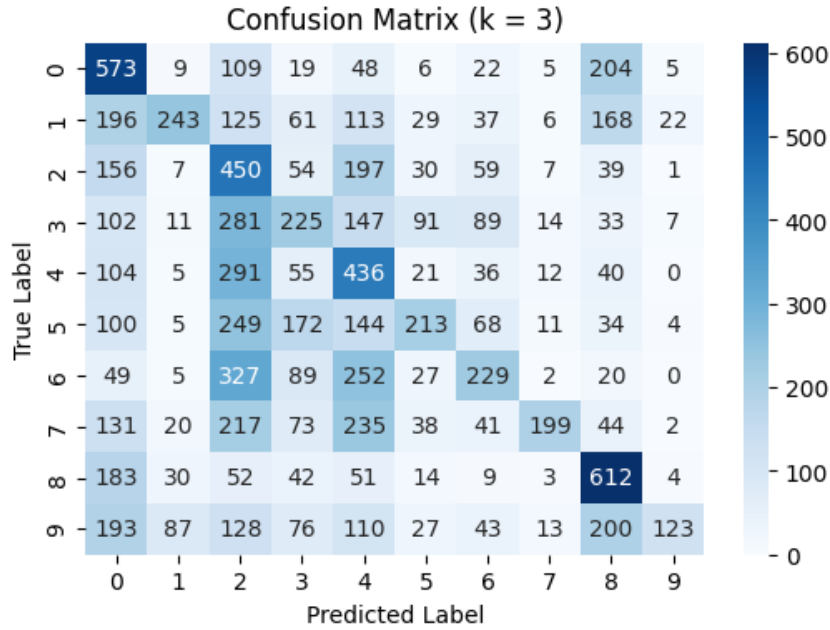


k ίσο με 3

Accuracy Score 0.3303

Class	Precision	Recall	F1-Score	Support
Airplane	0.32	0.57	0.41	1000
Automobile	0.58	0.24	0.34	1000
Bird	0.20	0.45	0.28	1000
Cat	0.26	0.23	0.24	1000
Deer	0.25	0.44	0.32	1000
Dog	0.43	0.21	0.28	1000
Frog	0.36	0.23	0.28	1000
Horse	0.73	0.20	0.31	1000
Ship	0.44	0.61	0.51	1000
Truck	0.73	0.12	0.21	1000
Accuracy			0.33	10000
Macro Avg	0.43	0.33	0.32	10000
Weighted Avg	0.43	0.33	0.32	10000

Table 2: Classification Report



Υλοποίηση Nearest Centroid με Scikit-Learn

1. **Υπολογισμός κεντροειδών:** Για κάθε κατηγορία k με δείγματα X_k , υπολογίζεται ο κεντροειδής της κατηγορίας ως το μέσο όλων των δειγμάτων στην κατηγορία:

$$C_k = \frac{1}{n_k} \sum_{x \in X_k} x,$$

όπου n_k είναι ο αριθμός των δειγμάτων στην κατηγορία k και x είναι τα χαρακτηριστικά των δειγμάτων.

2. **Υπολογισμός αποστάσεων:** Για ένα νέο δείγμα x_{test} , υπολογίζεται η απόσταση από τον κεντροειδή κάθε κατηγορίας C_k . Συνήθως χρησιμοποιείται η Ευκλείδεια απόσταση, η οποία δίνεται από τη σχέση:

$$d(x_{\text{test}}, C_k) = \|x_{\text{test}} - C_k\|.$$

3. **Ανάθεση κατηγορίας:** Το δείγμα x_{test} ταξινομείται στην κατηγορία της οποίας ο κεντροειδής είναι ο πλησιέστερος, δηλαδή στην κατηγορία k^* όπου:

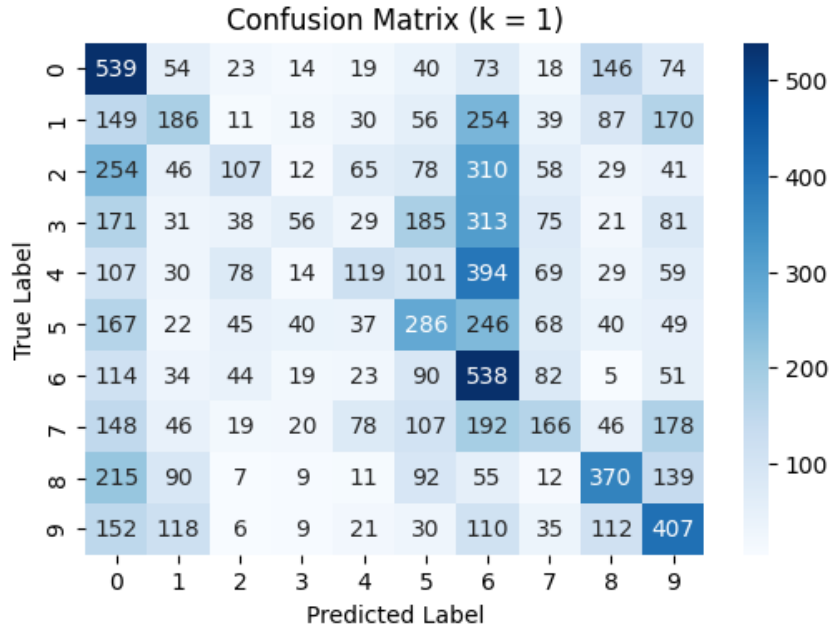
$$k^* = \arg \min_k d(x_{\text{test}}, C_k).$$

4. **Επιστροφή κατηγορίας:** Ο αλγόριθμος επιστρέφει την κατηγορία k^* ως την προβλεπόμενη κατηγορία για το δείγμα x_{test} .

Για τον αλγόριθμο nearest centroid έγινε μια προσπάθεια υλοποίησης του από το μηδέν, ωστόσο δεν χρησιμοποιήθηκε στις παρακάτω δοκιμές.

Class	Precision	Recall	F1-Score	Support
Airplane	0.27	0.54	0.36	1000
Automobile	0.28	0.19	0.22	1000
Bird	0.28	0.11	0.16	1000
Cat	0.27	0.06	0.09	1000
Deer	0.28	0.12	0.17	1000
Dog	0.27	0.29	0.28	1000
Frog	0.22	0.54	0.31	1000
Horse	0.27	0.17	0.20	1000
Ship	0.42	0.37	0.39	1000
Truck	0.33	0.41	0.36	1000
Accuracy		0.28		10000
Macro Avg	0.29	0.28	0.25	10000
Weighted Avg	0.29	0.28	0.25	10000

Table 3: Classification Report



Accuracy Score 0.2774

Παραλλαγές και Τεχνικές Βελτίωσης της Απόδοσης

Edge Detection

Η ανίχνευση ακμών, ή αλλιώς edge detection, είναι μια θεμελιώδης τεχνική επεξεργασίας εικόνας για την αναγνώριση και τον εντοπισμό των ορίων ή των ακμών των αντικειμένων σε μια εικόνα. Χρησιμοποιείται για να εντοπίσει και να ανιχνεύσει τις ασυνέχειες στην ένταση της εικόνας (image intensity) και να εξάγει τα περιγράμματα των αντικειμένων που υπάρχουν σε μια εικόνα. Σε προβλήματα image classification όπως αυτό, ενδέχεται να βελτιώσει την απόδοση των μοντέλων, καθώς βοηθά στην εξαγωγή των βασικών χαρακτηριστικών της εικόνας που καθορίζουν τη μορφή και τη δομή των αντικειμένων.

Sobel

1. **Φιλτράρισμα της εικόνας:** Η εικόνα μετατρέπεται σε gray scale, αν δεν είναι ήδη. Έπειτα ρησιμοποιούνται δύο φίλτρα (ή μάσκες) Sobel — ένα για την κατεύθυνση του οριζόντιου άξονα (G_x) και ένα για την κατακόρυφη κατεύθυνση (G_y). Αυτά τα φίλτρα εφαρμόζονται σε κάθε pixel της εικόνας.
2. **Υπολογισμός της κλίσης:** Μετά την εφαρμογή των φίλτρων, ο αλγόριθμος υπολογίζει την ένταση της ακμής σε κάθε σημείο της εικόνας, συνδυάζοντας τις πληροφορίες και από τις δύο κατευθύνσεις, χρησιμοποιώντας τον τύπο:

$$G = \sqrt{G_x^2 + G_y^2}$$

όπου G είναι η ένταση της ακμής και G_x και G_y οι τιμές των φίλτρων στην οριζόντια και κατακόρυφη κατεύθυνση αντίστοιχα.

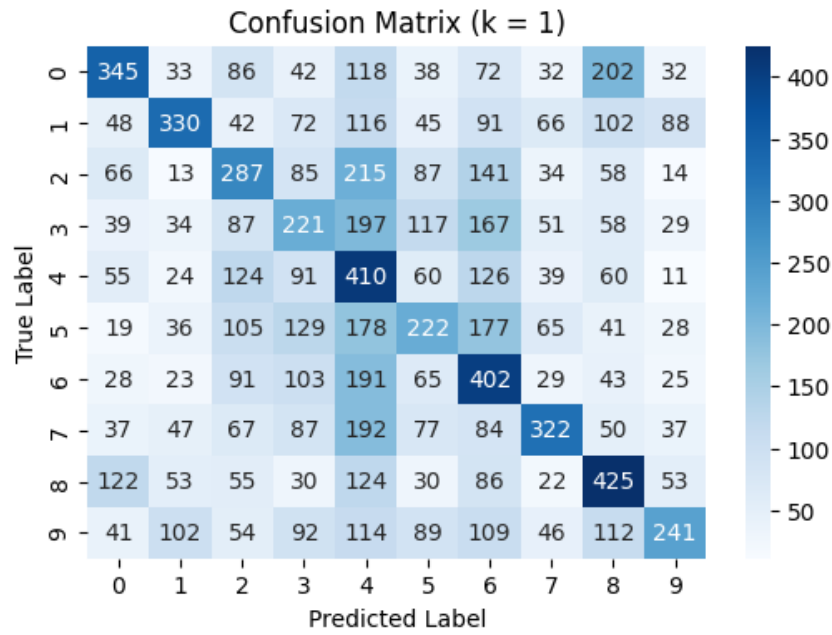
3. **Ανίχνευση ακμών:** Τα σημεία με υψηλή τιμή έντασης (δηλαδή τα σημεία με απότομες αλλαγές στην φωτεινότητα) θεωρούνται ως ακμές στην εικόνα.

Για την υλοποίηση του Sobel σε Python χρησιμοποιήθηκε η βιβλιοθήκη OpenCV και συγκεκριμένα οι εντολές `cvtColor` για την μετατροπή της εικόνας σε greyscale, η εντολή `GaussianBlur` για μείωση του θορύβου, η εντολή `Sobel` για το φιλτράρισμα και η εντολή `magnitude` για τον υπολογισμό του maginitude of the gradient.

kNN για k ίσο με 1 με Sobel Accuracy Score 0.3205

Class	Precision	Recall	F1-Score	Support
Airplane	0.43	0.34	0.38	1000
Automobile	0.47	0.33	0.39	1000
Bird	0.29	0.29	0.29	1000
Cat	0.23	0.22	0.23	1000
Deer	0.22	0.41	0.29	1000
Dog	0.27	0.22	0.24	1000
Frog	0.28	0.40	0.33	1000
Horse	0.46	0.32	0.38	1000
Ship	0.37	0.42	0.40	1000
Truck	0.43	0.24	0.31	1000
Accuracy		0.32		10000
Macro Avg	0.34	0.32	0.32	10000
Weighted Avg	0.34	0.32	0.32	10000

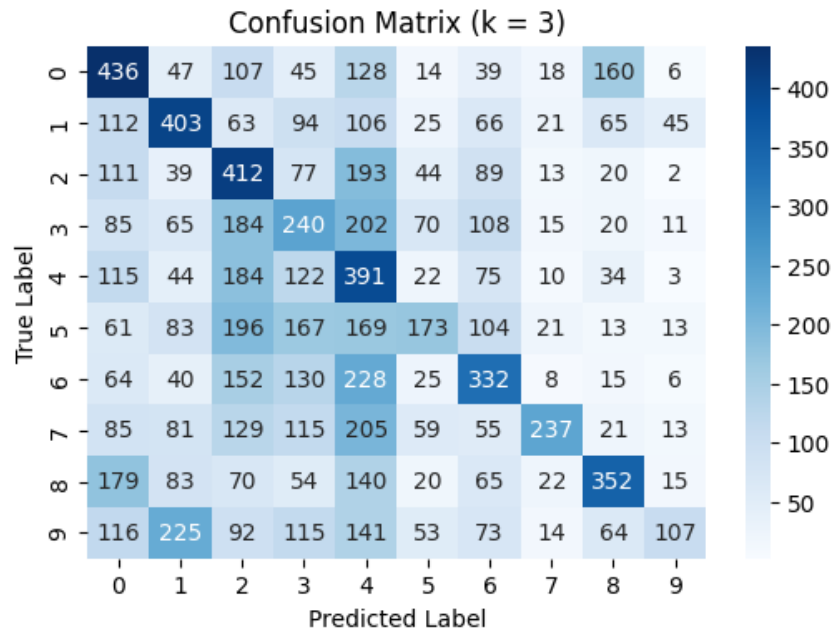
Table 4: Classification Report



kNN για k ίσο με 3 με Sobel Accuracy Score 0.3083

Class	Precision	Recall	F1-Score	Support
Airplane	0.32	0.44	0.37	1000
Automobile	0.36	0.40	0.38	1000
Bird	0.26	0.41	0.32	1000
Cat	0.21	0.24	0.22	1000
Deer	0.21	0.39	0.27	1000
Dog	0.34	0.17	0.23	1000
Frog	0.33	0.33	0.33	1000
Horse	0.63	0.24	0.34	1000
Ship	0.46	0.35	0.40	1000
Truck	0.48	0.11	0.18	1000
Accuracy		0.31		10000
Macro Avg	0.36	0.31	0.30	10000
Weighted Avg	0.36	0.31	0.30	10000

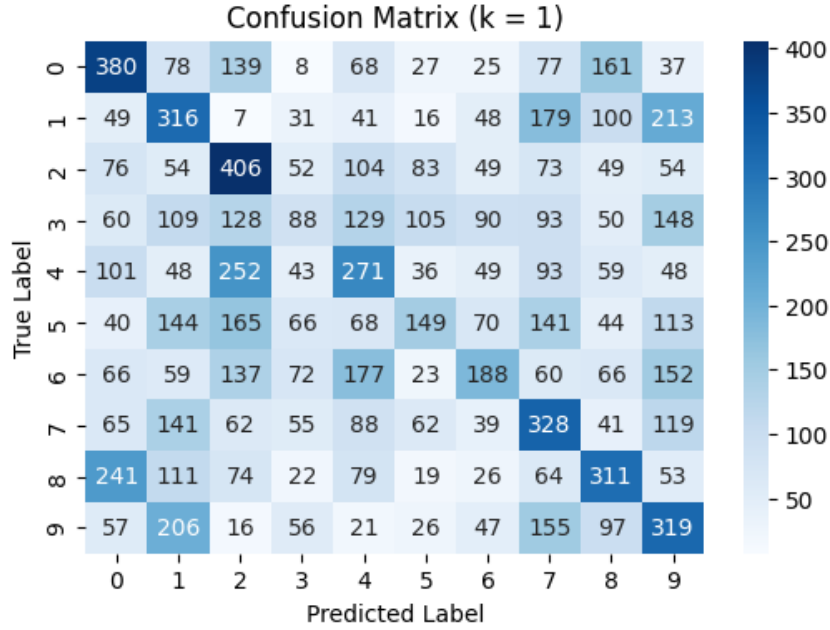
Table 5: Classification Report



Nearest Centroid Classifier $\mu\epsilon$ Sobel Accuracy Score is 0.2756

Class	Precision	Recall	F1-Score	Support
Airplane	0.33	0.38	0.36	1000
Automobile	0.25	0.32	0.28	1000
Bird	0.29	0.41	0.34	1000
Cat	0.18	0.09	0.12	1000
Deer	0.26	0.27	0.26	1000
Dog	0.27	0.15	0.19	1000
Frog	0.30	0.19	0.23	1000
Horse	0.26	0.33	0.29	1000
Ship	0.32	0.31	0.31	1000
Truck	0.25	0.32	0.28	1000
Accuracy		0.28		10000
Macro Avg	0.27	0.28	0.27	10000
Weighted Avg	0.27	0.28	0.27	10000

Table 6: Classification Report



Canny

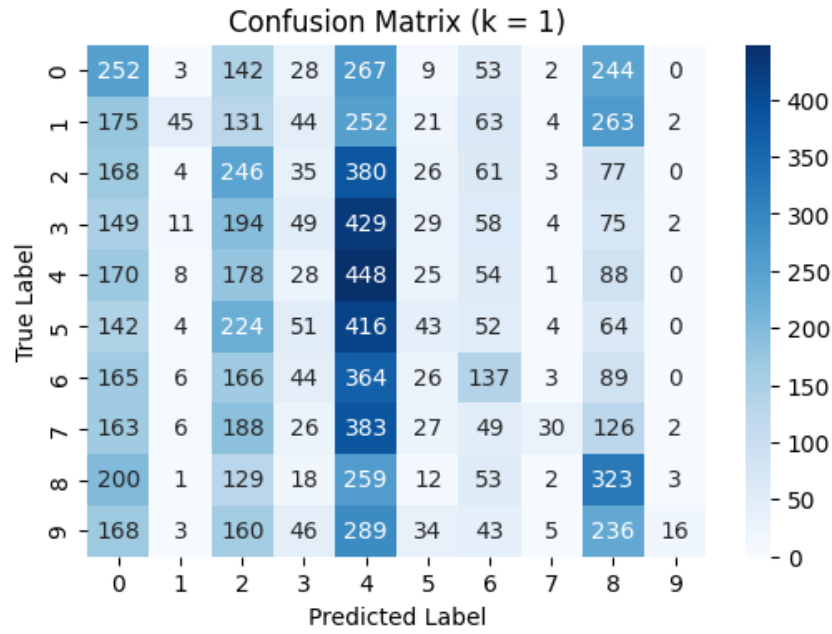
1. **Εφαρμογή Gaussian φίλτρου:** Η εικόνα υποβάλλεται πρώτα σε φιλτράρισμα με το Gaussian φίλτρο για να μειωθεί ο θόρυβος, διότι ο θόρυβος μπορεί να προκαλέσει την ανίχνευση λανθασμένων ακμών.
2. **Υπολογισμός των παραγώγων πρώτης τάξης:** Με τη χρήση των φίλτρων Sobel, υπολογίζονται τα παραγώγα πρώτης τάξης (δηλαδή, οι κλίσεις) για να εντοπιστούν οι περιοχές με μεγάλες αλλαγές στην ένταση (δηλαδή οι πιθανοί υποψήφιοι ακμές).
3. **Υπολογισμός της κατεύθυνσης των ακμών:** Η κατεύθυνση των ακμών υπολογίζεται χρησιμοποιώντας τα παραγώγα. Η κατεύθυνση αυτή είναι χρήσιμη για την ακριβή αναγνώριση των πραγματικών ακμών στην εικόνα.
4. **Αδύναμο και ισχυρό κατώφλι:** Στο επόμενο βήμα, εφαρμόζεται μια διαδικασία κατωφλίου για να αποφασιστεί ποια σημεία της εικόνας θα θεωρηθούν ακμές. Ο αλγόριθμος χρησιμοποιεί δύο κατώφλια: ένα για τις ισχυρές ακμές και ένα για τις αδύναμες ακμές. Οι αδύναμες ακμές συνδέονται με τις ισχυρές ακμές εάν βρίσκονται σε κοντινές θέσεις, ενώ οι υπόλοιπες απορρίπτονται.
5. **Τελική αναγνώριση ακμών:** Μετά τη διαδικασία της αφαίρεσης των αδύναμων και μη συνδεδεμένων ακμών, απομένουν μόνο οι ισχυρές και αξιόπιστες ακμές, οι οποίες τελικά θεωρούνται ως οι πραγματικές ακμές της εικόνας.

Για την υλοποίηση του Canny σε Python χρησιμοποιήθηκε και πάλι η βιβλιοθήκη OpenCV και συγκεκριμένα οι εντολές `cvtColor`, `GaussianBlur` και `Canny`.

kNN για k ίσο με 1 με Canny Accuracy Score of 0.1589

Class	Precision	Recall	F1-Score	Support
Airplane	0.14	0.25	0.18	1000
Automobile	0.49	0.04	0.08	1000
Bird	0.14	0.25	0.18	1000
Cat	0.13	0.05	0.07	1000
Deer	0.13	0.45	0.20	1000
Dog	0.17	0.04	0.07	1000
Frog	0.22	0.14	0.17	1000
Horse	0.52	0.03	0.06	1000
Ship	0.20	0.32	0.25	1000
Truck	0.64	0.02	0.03	1000
Accuracy		0.16		10000
Macro Avg	0.28	0.16	0.13	10000
Weighted Avg	0.28	0.16	0.13	10000

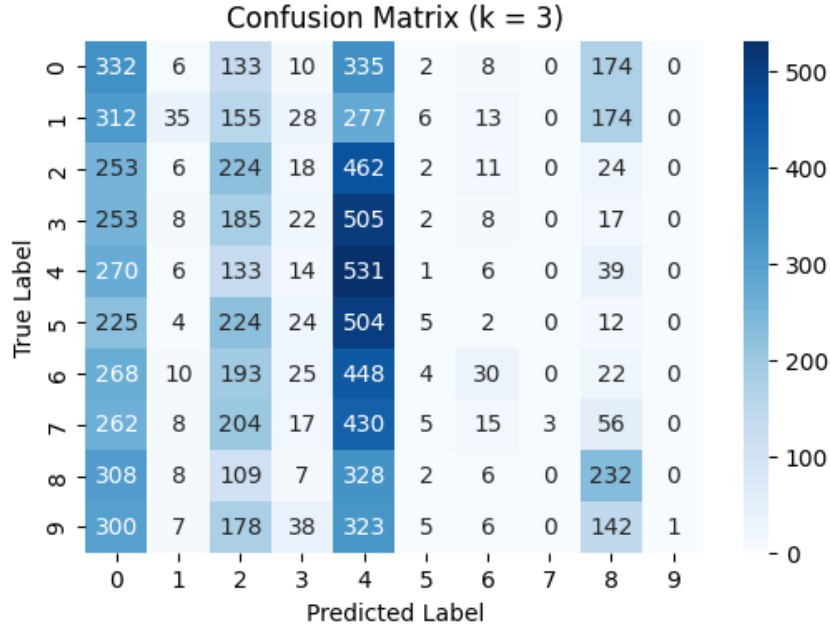
Table 7: Classification Report



kNN για k ίσο με 3 με Canny Accuracy score of 0.1415

Class	Precision	Recall	F1-Score	Support
Airplane	0.12	0.33	0.18	1000
Automobile	0.36	0.04	0.06	1000
Bird	0.13	0.22	0.16	1000
Cat	0.11	0.02	0.04	1000
Deer	0.13	0.53	0.21	1000
Dog	0.15	0.01	0.01	1000
Frog	0.29	0.03	0.05	1000
Horse	1.00	0.00	0.01	1000
Ship	0.26	0.23	0.25	1000
Truck	1.00	0.00	0.00	1000
Accuracy		0.14		10000
Macro Avg	0.35	0.14	0.10	10000
Weighted Avg	0.35	0.14	0.10	10000

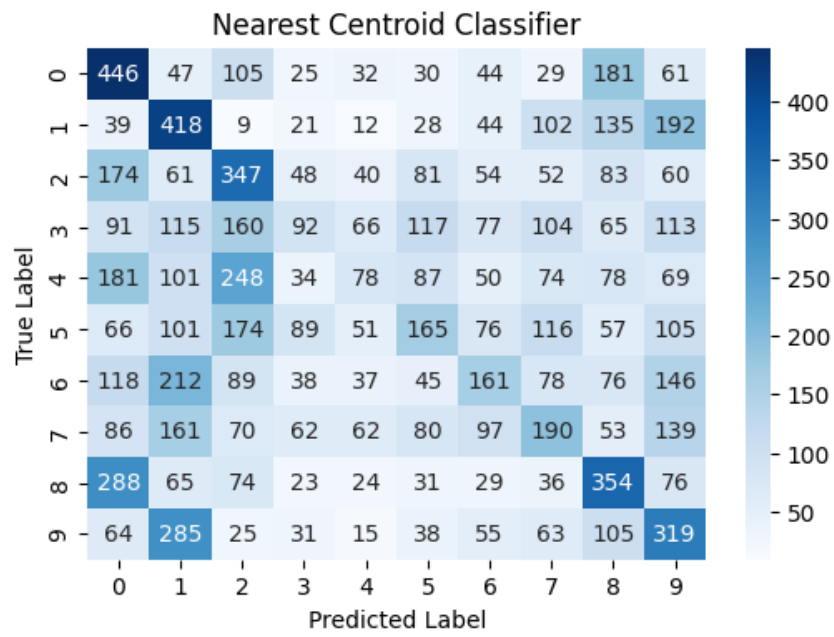
Table 8: Classification Report



Nearest Centroid Classifier $\mu\epsilon$ Canny Accuracy score of 0.257

Class	Precision	Recall	F1-Score	Support
Airplane	0.29	0.45	0.35	1000
Automobile	0.27	0.42	0.33	1000
Bird	0.27	0.35	0.30	1000
Cat	0.20	0.09	0.13	1000
Deer	0.19	0.08	0.11	1000
Dog	0.24	0.17	0.19	1000
Frog	0.23	0.16	0.19	1000
Horse	0.23	0.19	0.21	1000
Ship	0.30	0.35	0.32	1000
Truck	0.25	0.32	0.28	1000
Accuracy		0.26		10000
Macro Avg	0.24	0.26	0.24	10000
Weighted Avg	0.24	0.26	0.24	10000

Table 9: Classification Report



Color Space Transformation

Στην προσπάθεια για βελτίωση της απόδοσης πραγματοποιήθηκε περαιτέρω έρευνα η οποία έδειξε ότι για αποτελεσματικό (*image classification*), συχνά πραγματοποιείται *color space transformation*, καθώς η αναπαράσταση μιας εικόνας σε διαφορετικούς χρωματικούς χώρους μπορεί να αναδείξει χαρακτηριστικά που είναι λιγότερο εμφανή στο RGB. Συγκεκριμένα, οι χρωματικοί χώροι που δοκιμάστηκαν είναι οι HSV, LAB, YCbCr.

Σημειώνεται ότι, καθώς οι μέθοδοι HSV και YCbCr δεν παρήγαγαν ικανοποιητικά αποτελέσματα, και δεδομένης της έκτασης της παρούσας μελέτης, παρουσιάζονται αναλυτικά μόνο τα αποτελέσματα για τη μέθοδο LAB.

Ωστόσο, ο κώδικας για τις μεθόδους HSV και YCbCr περιλαμβάνεται, ώστε να είναι δυνατή η περαιτέρω ανάλυση και η εξαγωγή επιπρόσθετων μετρικών, όπως το *confusion matrix* και το *classification report*, εάν αυτό κρίνεται απαραίτητο.

HSV

Ο HSV (Hue, Saturation, Value) διαχωρίζει την απόχρωση από τη φωτεινότητα, επιτρέποντας καλύτερη ανίχνευση αντικειμένων ανεξάρτητα από τον φωτισμό.

YCbCr

Ο YCbCr (Luminance, Blue-difference, Red-difference) διαχωρίζει τη φωτεινότητα από τις πληροφορίες χρώματος, βελτιώνοντας την απόδοση σε συνθήκες χαμηλού φωτισμού και μειώνοντας τον θόρυβο από τα χρώματα.

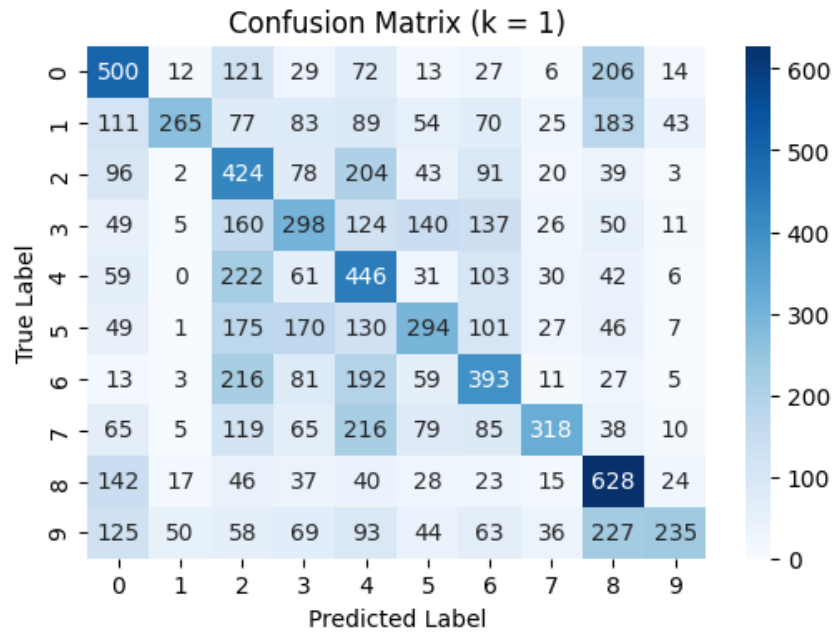
LAB

Το LAB (Lightness, A, B) επικεντρώνεται στην αντίληψη του φωτός και του χρώματος όπως ορίζονται από τον ανθρώπινο οπτικό σύστημα, καθιστώντας τις πληροφορίες χρώματος περισσότερο διαχωρίσιμες.

kNN για k ίσο με 1 Accuracy Score of 0.3801

Class	Precision	Recall	F1-Score	Support
Airplane	0.41	0.50	0.45	1000
Automobile	0.74	0.27	0.39	1000
Bird	0.26	0.42	0.32	1000
Cat	0.31	0.30	0.30	1000
Deer	0.28	0.45	0.34	1000
Dog	0.37	0.29	0.33	1000
Frog	0.36	0.39	0.38	1000
Horse	0.62	0.32	0.42	1000
Ship	0.42	0.63	0.51	1000
Truck	0.66	0.23	0.35	1000
Accuracy		0.38		10000
Macro Avg	0.44	0.38	0.38	10000
Weighted Avg	0.44	0.38	0.38	10000

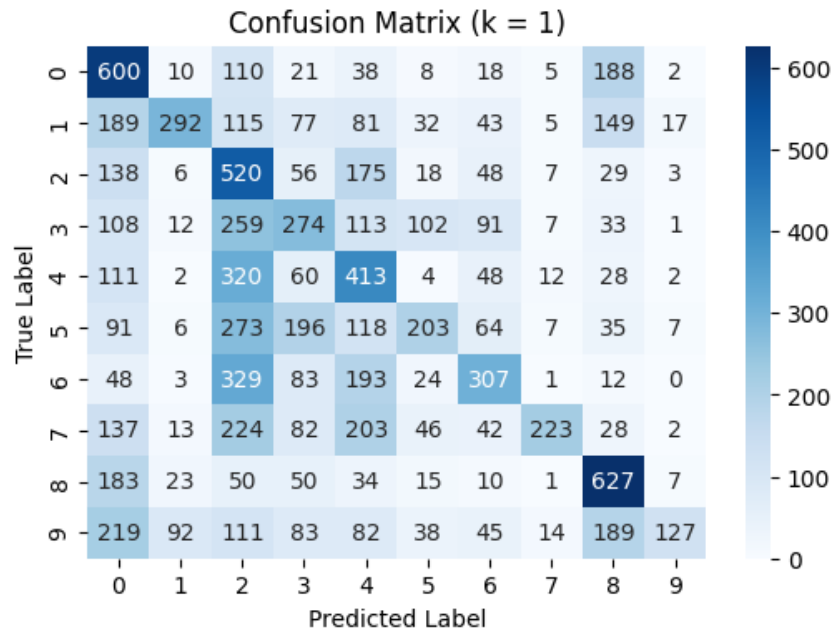
Table 10: Classification Report



kNN για k ίσο με 3 Accuracy Score of 0.3586

Class	Precision	Recall	F1-Score	Support
Airplane	0.33	0.60	0.42	1000
Automobile	0.64	0.29	0.40	1000
Bird	0.23	0.52	0.31	1000
Cat	0.28	0.27	0.28	1000
Deer	0.28	0.41	0.34	1000
Dog	0.41	0.20	0.27	1000
Frog	0.43	0.31	0.36	1000
Horse	0.79	0.22	0.35	1000
Ship	0.48	0.63	0.54	1000
Truck	0.76	0.13	0.22	1000
Accuracy		0.36		10000
Macro Avg	0.46	0.36	0.35	10000
Weighted Avg	0.46	0.36	0.35	10000

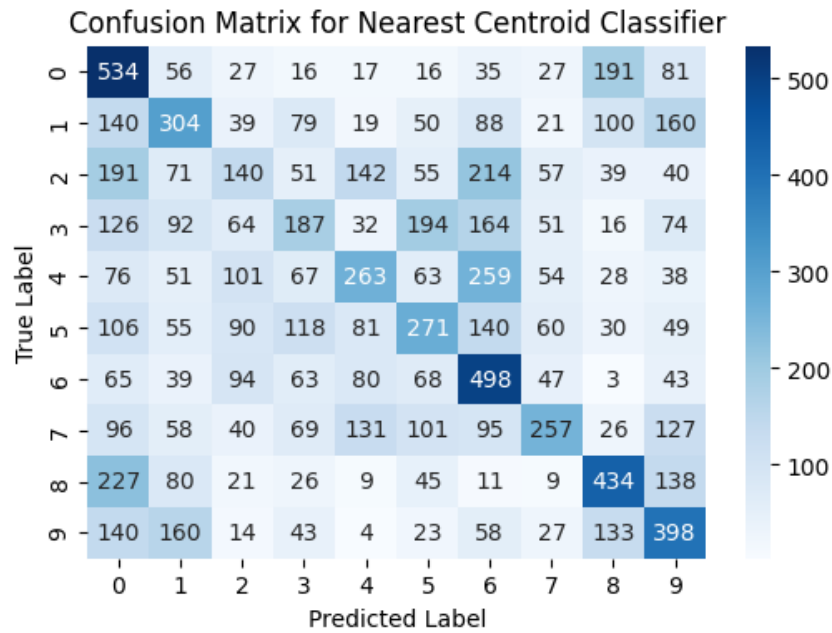
Table 11: Classification Report



Nearest Centroid Classifier Accuracy Score of 0.3286

Class	Precision	Recall	F1-Score	Support
Airplane	0.31	0.53	0.40	1000
Automobile	0.31	0.30	0.31	1000
Bird	0.22	0.14	0.17	1000
Cat	0.26	0.19	0.22	1000
Deer	0.34	0.26	0.30	1000
Dog	0.31	0.27	0.29	1000
Frog	0.32	0.50	0.39	1000
Horse	0.42	0.26	0.32	1000
Ship	0.43	0.43	0.43	1000
Truck	0.35	0.40	0.37	1000
Accuracy		0.33		10000
Macro Avg	0.33	0.33	0.32	10000
Weighted Avg	0.33	0.33	0.32	10000

Table 12: Classification Report



Distance Metric

Αφού ολοκληρωθεί το `reshape` των δεδομένων, η αναπαράσταση του διανύσματος είναι η εξής:

$$a = [\text{pixel}_1\text{-}R, \text{pixel}_1\text{-}G, \text{pixel}_1\text{-}B, \text{pixel}_2\text{-}R, \text{pixel}_2\text{-}G, \text{pixel}_2\text{-}B, \dots, \text{pixel}_{1024}\text{-}R, \text{pixel}_{1024}\text{-}G, \text{pixel}_{1024}\text{-}B]$$

Με τον τρόπο αυτό, το διανύσμα γίνεται μονοδιάστατο, με 1024 τριπλέτες αριθμών, όπου κάθε τριπλέτα αντιστοιχεί σε ένα pixel της εικόνας. Κάθε τριπλέτα περιλαμβάνει μία τιμή για το κόκκινο (R), μία για το πράσινο (G) και μία για το μπλε (B). Αρχικά, οι τιμές αυτές κυμαίνονται στο εύρος 0-255, όμως μετά την κανονικοποίηση, ανήκουν στο εύρος 0-1.

Ο αλγόριθμος kNN (k-Nearest Neighbors) βασίζεται στον υπολογισμό αποστάσεων μεταξύ των διανυσμάτων χαρακτηριστικών των εικόνων. Μέχρι στιγμής, χρησιμοποιούνταν η ευκλείδεια απόσταση, η οποία ορίζεται ως εξής:

Αν θεωρήσουμε δύο διανύσματα, a και b , που αντιπροσωπεύουν τα χαρακτηριστικά δύο εικόνων, τα οποία έχουν τις ακόλουθες μορφές:

$$a = [a_1, a_2, a_3, \dots, a_{3072}]$$

$$b = [b_1, b_2, b_3, \dots, b_{3072}]$$

τότε η ευκλείδεια απόσταση $d(a, b)$ μεταξύ αυτών των δύο διανυσμάτων υπολογίζεται ως εξής:

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_{3072} - b_{3072})^2}$$

Ωστόσο υπάρχουν και άλλες αποστάσεις που μπορούν να χρησιμοποιηθούν όπως cosine similarity, minkowski, chebyhev. Αφού δοκιμάστηκαν προέκυψε πως τα καλύτερα αποτελέσματα λαμβάνονται για cosine similarity. Το minkowski είχε παρόμοια αποτελέσματα αλλά λίγο χαμηλότερα ενώ το chebyshev είχε με διαφορά τα χειρότερα αφού το accuracy score του ήταν κοντά στο 17 με 18%.

Cosine Similarity

Η απόσταση αυτή για δύο διανύσματα ορίζεται ως εξής

$$\text{cosine_distance}(A, B) = 1 - \frac{A \cdot B}{\|A\| \cdot \|B\|}$$

Η cosine similarity είναι πιο κατάλληλη από την ευκλείδεια απόσταση σε περιπτώσεις ταξινόμησης εικόνας, καθώς εστιάζει στη γωνιακή ομοιότητα μεταξύ δύο διανυσμάτων.

Principal Component Analysis - PCA

Το Principal Component Analysis - PCA είναι μια τεχνική μείωσης διαστάσεων που χρησιμοποιείται για να συμπυκνώσει την πληροφορία ενός συνόλου δεδομένων, διατηρώντας παράλληλα όσο το δυνατόν περισσότερη από τη σημαντική πληροφορία.

Βέλτιστος Συνδυασμός

Με την εφαρμογή τόσο της μεθόδου PCA όσο και της Cosine Similarity παρατηρήθηκε σημαντική βελτίωση στην απόδοση και των δύο αλγορίθμων ταξινόμησης. Για λόγους έκτασης της παρούσας εργασίας, παρουσιάζεται απευθείας ο βέλτιστος συνδυασμός παραμέτρων.

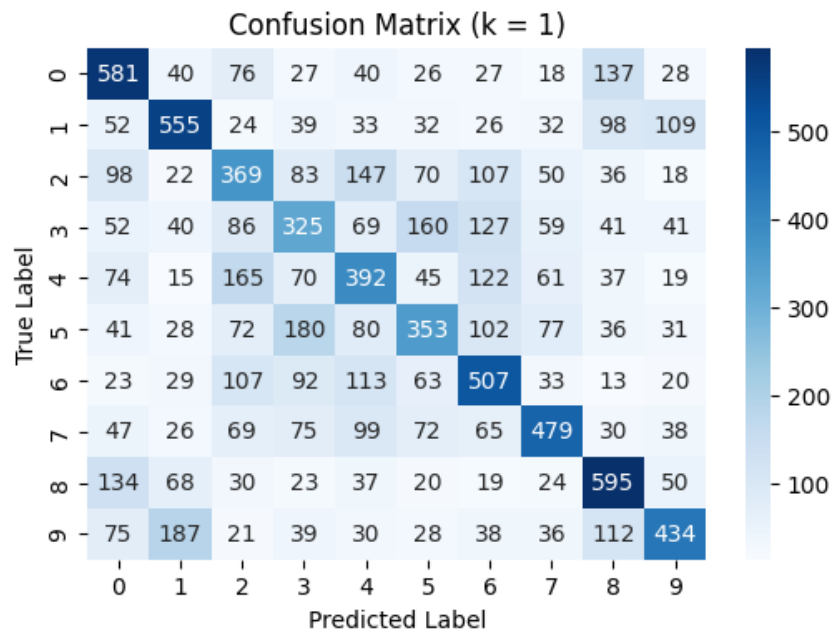
Ο υψηλότερος συντελεστής απόδοσης (Accuracy Score) επιτεύχθηκε με τη χρήση του χρωματικού χώρου *LAB*, την Cosine Similarity, και για $k = 1$ με εφαρμογή της μεθόδου PCA. Τα αποτελέσματα παρουσιάζονται στις ακόλουθες εικόνες.

Σημειώνεται ότι το PCA πρέπει να εφαρμοστεί μετά από την μετατροπή της εικόνας στο διαφορετικό Color Space. Αυτό συμβαίνει γιατί μετά τον μετασχηματισμό, χαρακτηριστικά της εικόνας τα οποία μπορεί πριν να μην ήταν σημαντικά πλέον να έχουν γίνει. Έτσι είναι σημαντικό να μην έχει χαθεί πληροφορία πριν τον μετασχηματισμό για να διατηρηθούν τα πλεονεκτήματα του νέου χρωματικού χώρου πριν αποφασίσει το PCA ποια δεδομένα θα κρατήσει.

kNN για k ίσο με 1 Accuracy Score of 0.4601

Class	Precision	Recall	F1-Score	Support
Airplane	0.49	0.58	0.53	1000
Automobile	0.55	0.56	0.55	1000
Bird	0.36	0.37	0.37	1000
Cat	0.34	0.33	0.33	1000
Deer	0.38	0.39	0.38	1000
Dog	0.41	0.35	0.38	1000
Frog	0.44	0.51	0.47	1000
Horse	0.55	0.48	0.51	1000
Ship	0.52	0.59	0.56	1000
Truck	0.55	0.43	0.49	1000
Accuracy		0.46		10000
Macro Avg	0.46	0.46	0.46	10000
Weighted Avg	0.46	0.46	0.46	10000

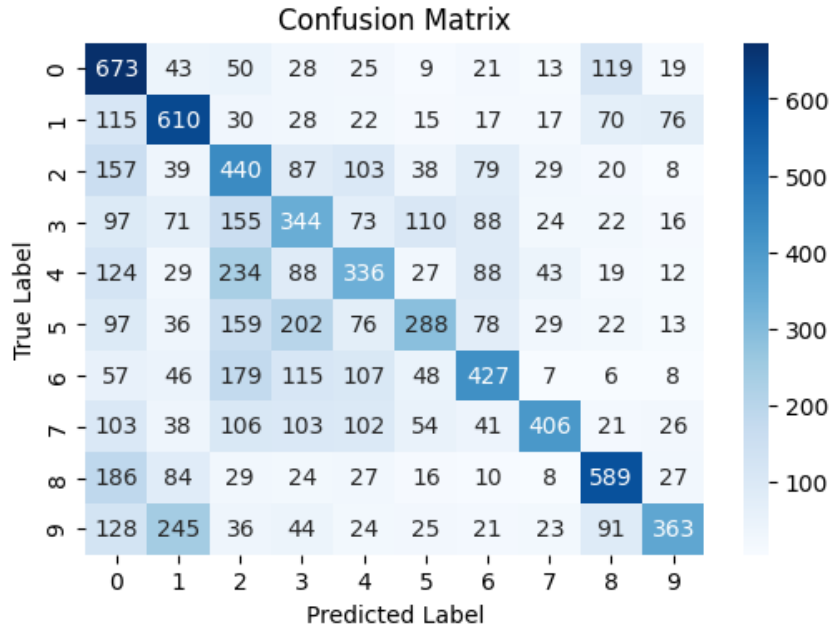
Table 13: Classification Report



kNN για k ίσο με 3 Accuracy Score of 0.4476

Class	Precision	Recall	F1-Score	Support
Airplane	0.39	0.67	0.49	1000
Automobile	0.49	0.61	0.54	1000
Bird	0.31	0.44	0.36	1000
Cat	0.32	0.34	0.33	1000
Deer	0.38	0.34	0.35	1000
Dog	0.46	0.29	0.35	1000
Frog	0.49	0.43	0.46	1000
Horse	0.68	0.41	0.51	1000
Ship	0.60	0.59	0.60	1000
Truck	0.64	0.36	0.46	1000
Accuracy		0.45		10000
Macro Avg	0.48	0.45	0.45	10000
Weighted Avg	0.48	0.45	0.45	10000

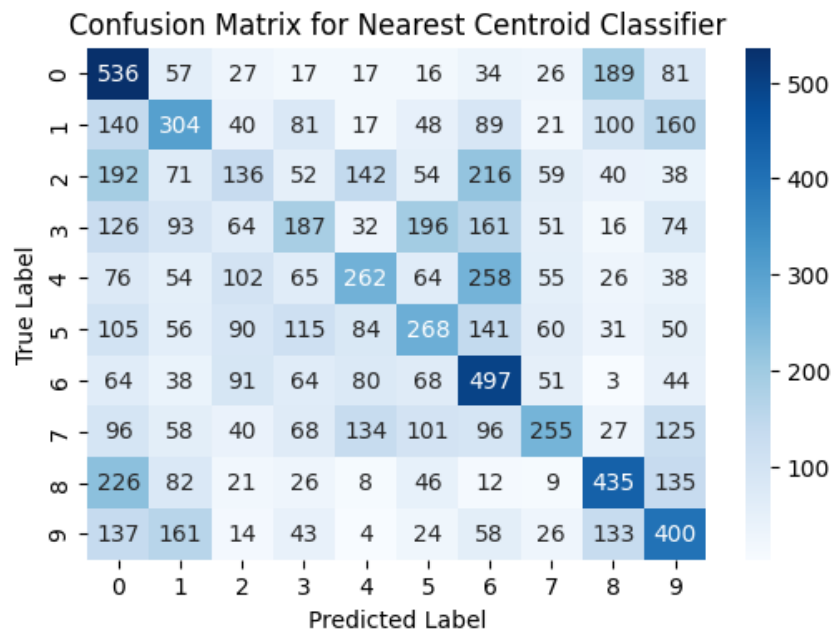
Table 14: Classification Report



Nearest Centroid Classifier Accuracy Score of 0.328

Class	Precision	Recall	F1-Score	Support
Airplane	0.32	0.54	0.40	1000
Automobile	0.31	0.30	0.31	1000
Bird	0.22	0.14	0.17	1000
Cat	0.26	0.19	0.22	1000
Deer	0.34	0.26	0.29	1000
Dog	0.30	0.27	0.28	1000
Frog	0.32	0.50	0.39	1000
Horse	0.42	0.26	0.32	1000
Ship	0.43	0.43	0.43	1000
Truck	0.35	0.40	0.37	1000
Accuracy		0.33		10000
Macro Avg	0.33	0.33	0.32	10000
Weighted Avg	0.33	0.33	0.32	10000

Table 15: Classification Report



Τελικά το βέλτιστο μοντέλο προέκυψε ως εξής (το σχήμα αναφέρεται στον δείκτη accuracy score)

