



ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ
Εφαρμογή Ιστού για την ανίχνευση πλαστών
υπογραφών

Του φοιτητή
Τσιούμας Δημήτρης
Αρ. Μητρώου: 185300

Επιβλέπων
Στέφανος Ουγιάρογλου
Επίκουρος Καθηγητής

20 Σεπτεμβρίου 2026

Εφαρμογή Ιστού για την ανίχνευση πλαστών υπογραφών

Κωδικός Π. Ε. 24269

Ονοματεπώνυμο φοιτητή Τσιούμας Δημήτρης

Ονοματεπώνυμο εισηγητή Ουγιάρογλου Στέφανος

Ημερομηνία ανάληψης Π. Ε. 22-10-2024

Ημερομηνία περάτωσης Π. Ε. 20-09-2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ. ΠΑ. Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Τσιούμα Δημήτρη που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Αφιέρωση

Η παρούσα εργασία αφιερώνεται στον καθηγητή μου, Στέφανο Ουγιάρογλου, για την πολύτιμη καθοδήγηση, τη συνεχή υποστήριξη και την ευκαιρία που μου έδωσε να συνεργαστώ μαζί του στο πλαίσιο της παρούσας μελέτης.

Πρόλογος

Η επιλογή της συγκεκριμένης πτυχιακής εργασίας προέκυψε από το ενδιαφέρον μου για τη **Μηχανική Μάθηση** και τις εφαρμογές της σε πραγματικά προβλήματα, και ειδικότερα για τη δυνατότητα αξιοποίησης τεχνικών Μηχανικής και Βαθιάς Μάθησης στην επαλήθευση χειρόγραφων υπογραφών. Ιδιαίτερο κίνητρο αποτέλεσε η προοπτική εξέλιξης μιας προηγούμενης ερευνητικής προσέγγισης σε μία ολοκληρωμένη και εύχρηστη διαδικτυακή εφαρμογή, η οποία θα μπορούσε να αξιοποιηθεί και από μη εξειδικευμένους χρήστες.

Μέσα από την ανάπτυξη της **Signalize** είχα την ευκαιρία να συνδυάσω γνώσεις από διαφορετικά πεδία, όπως η ανάπτυξη web εφαρμογών, η σχεδίαση βάσεων δεδομένων, η επεξεργασία εικόνας και η Μηχανική Μάθηση. Παράλληλα, η εργασία με βοήθησε να αποκτήσω μεγαλύτερη εξοικείωση με την πειραματική αξιολόγηση μοντέλων, την επιλογή κατάλληλων τεχνολογιών και την ενσωμάτωση διαφορετικών υποσυστημάτων σε μία ενιαία εφαρμογή. Συνολικά, η εκπόνηση της εργασίας αποτέλεσε μια ιδιαίτερα χρήσιμη εμπειρία, τόσο σε τεχνικό όσο και σε ερευνητικό επίπεδο.

Περίληψη

Η παρούσα πτυχιακή εργασία επικεντρώνεται στη σχεδίαση και ανάπτυξη της διαδικτυακής εφαρμογής **Signalize**, η οποία αξιοποιεί τεχνικές Τεχνητής Νοημοσύνης για την υποστήριξη της ανίχνευσης πλαστών χειρόγραφων υπογραφών. Η εργασία βασίζεται στα συμπεράσματα προηγούμενης πτυχιακής μελέτης, στην οποία διερευνήθηκε η αποτελεσματικότητα συνδυασμών Μηχανικής και Βαθιάς Μάθησης για τη διάκριση γνήσιων και πλαστών υπογραφών και αναδείχθηκε η σημαντική δυναμική των τεχνικών αυτών στο συγκεκριμένο πεδίο. Ωστόσο, η προηγούμενη προσέγγιση είχε κυρίως πειραματικό και ερευνητικό χαρακτήρα, χωρίς να συνοδεύεται από ένα ολοκληρωμένο περιβάλλον μέσω του οποίου ένας τελικός χρήστης να μπορεί να αξιοποιήσει εύκολα τα μοντέλα στην πράξη. Η **Signalize** επιχειρεί να καλύψει αυτό το κενό, προσφέροντας ένα εύχρηστο web περιβάλλον στο οποίο ο χρήστης μπορεί να δημιουργήσει λογαριασμό, να καταχωρίσει γνήσια δείγματα υπογραφών και στη συνέχεια να υποβάλει νέες υπογραφές προς αξιολόγηση, είτε μέσω υπάρχουσας εικόνας είτε μέσω άμεσης λήψης φωτογραφίας από κινητή συσκευή. Η εφαρμογή υποστηρίζει δύο διαφορετικούς τρόπους εκτίμησης της γνησιότητας, έναν βασισμένο σε συνδυασμό αποτελέσματος εκπαιδευμένου μοντέλου και βαθμού ομοιότητας με τα γνήσια δείγματα, με δυνατότητα ρύθμισης της σχετικής βαρύτητας των δύο συνιστωσών, και έναν δεύτερο μέσω υπηρεσίας Τεχνητής Νοημοσύνης της OpenAI. Παράλληλα, παρέχεται η δυνατότητα δημιουργίας πρόσθετων μελών, ώστε ο ίδιος λογαριασμός να μπορεί να διαχειρίζεται ξεχωριστά σύνολα γνήσιων υπογραφών και ελέγχων για περισσότερα πρόσωπα. Με τον τρόπο αυτό, η εργασία μεταφέρει τα αποτελέσματα της προηγούμενης ερευνητικής προσέγγισης σε ένα ολοκληρωμένο, λειτουργικό και προσανατολισμένο στον τελικό χρήστη πληροφοριακό σύστημα.

Abstract

This thesis presents the design and development of Signalyze, a web-based application that uses artificial intelligence techniques to support the detection of forged handwritten signatures. The project extends earlier research on machine- and deep-learning methods for offline signature verification by translating an experimental approach into a practical tool for non-expert users. Signalyze enables users to create an account, store genuine reference signatures, and evaluate new signatures by uploading an image or capturing one with a mobile device. Its primary verification mechanism combines the probability produced by a machine-learning classifier with the cosine similarity between the signature under examination and the genuine reference signatures stored by the user. An experimental study evaluated combinations of convolutional neural network feature extractors and machine-learning classifiers, from which DenseNet169 with Naive Bayes was selected as the best-performing configuration. A second experiment examined alternative weightings of the classifier probability and similarity score. The highest classification accuracy, 62.38%, was achieved when the classifier probability was weighted at 35% and the similarity score at 65%. The application also allows users to manage additional members under a single account and provides an optional alternative evaluation through the OpenAI API. The web application was developed using PHP and CodeIgniter 4, while the signature-analysis subsystem was implemented in Python with TensorFlow, Keras, scikit-learn, and OpenCV.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα καθηγητή μου, Στέφανο Ουγιάρογλου, για την πολύτιμη καθοδήγηση, τη συνεχή υποστήριξη και τις ουσιαστικές προτάσεις του καθ' όλη τη διάρκεια εκπόνησης της παρούσας πτυχιακής εργασίας.

Περιεχόμενα

Πρόλογος.....	iv
Περίληψη.....	v
Abstract	vi
Ευχαριστίες	vii
Περιεχόμενα	viii
Κατάλογος Σχημάτων	xi
Κατάλογος Πινάκων.....	xii
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Ταυτοπροσωπία και φυσικές υπογραφές.....	1
1.2 Το πρόβλημα της πλαστογράφησης υπογραφών.....	2
1.3 Ανίχνευση πλαστών υπογραφών μέσω TN	2
1.4 Κίνητρο	3
1.5 Συνεισφορά.....	3
1.6 Οργάνωση της εργασίας.....	4
Κεφάλαιο 2ο: Σχετικές εργασίες.....	6
2.1 Μοντέλα TN για ανίχνευση πλαστών υπογραφών	6
2.2 Εφαρμογές ανίχνευσης και επαλήθευσης υπογραφών	6
Κεφάλαιο 3ο: Πειραματική μελέτη.....	8
3.1 Σκοπός πειραματικής μελέτης.....	8
3.2 Σύνολο δεδομένων	8
3.3 Πειραματική εγκατάσταση.....	9
3.4 Πειραματικά αποτελέσματα	11
3.4.1 Αποτελέσματα αξιολόγησης μοντέλων	11
3.4.2 Αποτελέσματα αξιολόγησης βαρών	13
Κεφάλαιο 4ο: Γλώσσες και τεχνολογίες.....	14
4.1 Backend.....	14
4.2 Βάση δεδομένων	15
4.3 Frontend	15
4.4 Μηχανική Μάθηση και επεξεργασία υπογραφών.....	16
Κεφάλαιο 5ο: Σχεδίαση και Υλοποίηση της Signalyze	17
5.1 Λειτουργικές απαιτήσεις	17

5.2	Αρχιτεκτονική της Εφαρμογής (use cases)	18
5.2.1	Διάγραμμα περιπτώσεων χρήσης	18
5.2.2	Βασικές περιπτώσεις χρήσης.....	21
5.2.3	Περιγραφή κύριας περίπτωσης χρήσης	21
5.3	Σχεδίαση Βάσης Δεδομένων	22
5.3.1	Διάγραμμα Οντοτήτων-Συσχετίσεων	22
5.3.2	Σχεσιακό Σχήμα.....	23
5.4	Υλοποίηση του Backend	25
5.4.1	Δρομολόγηση Αιτημάτων.....	25
5.4.2	Αυθεντικοποίηση και έλεγχος πρόσβασης	26
5.4.3	Μεταφόρτωση και αποθήκευση υπογραφών.....	26
5.4.4	Διαχείριση γνήσιων υπογραφών αναφοράς.....	28
5.4.5	Έλεγχοι πριν από την ανάλυση.....	29
5.4.6	Επικοινωνία PHP και Python	29
5.4.7	Εξαγωγή χαρακτηριστικών και ταξινόμηση.....	31
5.4.8	Υπολογισμός ομοιότητας και τελικού βαθμού	33
5.4.9	Αποθήκευση του αποτελέσματος της ανάλυσης	34
5.4.10	Εναλλακτική αξιολόγηση μέσω OpenAI	35
5.4.11	Αποστολή και έλεγχος της απάντησης του OpenAI API.....	37
5.4.12	Συνολική ροή του Backend.....	38
5.5	Υλοποίηση του Frontend.....	40
5.5.1	Κοινό layout και δομή των views.....	40
5.5.2	Φόρμα εισαγωγής και επεξεργασίας υπογραφής.....	42
5.5.3	Επιλογή εικόνας από αρχείο ή κάμερα.....	43
5.5.4	Μαζική προσθήκη και προεπισκόπηση γνήσιων υπογραφών	43
5.5.5	Προβολή και φιλτράρισμα υπογραφών	45
5.5.6	Responsive σχεδίαση και ασφαλής εμφάνιση δεδομένων.....	47
Κεφάλαιο 6ο:	Παρουσίαση της Signalyze	49
6.1	Αρχική σελίδα	49
6.2	Σύνδεση χρήστη	49
6.3	Δημιουργία νέου λογαριασμού.....	50
6.4	Σελίδα πληροφοριών της εφαρμογής	50
6.5	Οδηγίες χρήσης	51
6.6	Προσθήκη νέας υπογραφής.....	52
6.7	Μαζική μεταφόρτωση γνήσιων υπογραφών	53

6.8	Προβολή γνήσιας υπογραφής αναφοράς.....	53
6.9	Υπογραφή πριν από την εκτέλεση της ανάλυσης.....	54
6.10	Προβολή αποτελεσμάτων ανάλυσης.....	55
6.11	Λίστα υπογραφών.....	55
6.12	Διαχείριση μελών	56
6.13	Προβολή προφίλ.....	57
6.14	Επεξεργασία προφίλ και παραμέτρων ανάλυσης	57
Κεφάλαιο 7ο:	Συμπεράσματα και Μελλοντικές Επεκτάσεις.....	58
7.1	Συμπεράσματα.....	58
7.2	Μελλοντικές Αναβαθμίσεις.....	59
ΒΙΒΛΙΟΓΡΑΦΙΑ.....		61

Κατάλογος Σχημάτων

Σχήμα 5.2.1: Διάγραμμα περιπτώσεων χρήσης.....	20
Σχήμα 5.3.1: Διάγραμμα ERD (Entity Relationship Diagram).....	23
Σχήμα 5.4.1: Βασικές διαδρομές για τη διαχείριση υπογραφών και την αυθεντικοποίηση.....	25
Σχήμα 5.4.2: Έλεγχος σύνδεσης και ιδιοκτησίας υπογραφής.....	26
Σχήμα 5.4.3.1: Τμήμα της μεθόδου create() για τον έλεγχο του αρχείου μιας νέας υπογραφής.....	27
Σχήμα 5.4.3.2: Αποθήκευση εικόνας και δημιουργία της εγγραφής Signature στη μέθοδο create()....	28
Σχήμα 5.4.4: Έλεγχος του μέγιστου αριθμού γνήσιων υπογραφών ανά πρόσωπο.....	28
Σχήμα 5.4.5: Έλεγχοι που προηγούνται της ανάλυσης μιας υπογραφής.....	29
Σχήμα 5.4.6.1: Δημιουργία εντολής εκτέλεσης και μεταβίβαση των παραμέτρων στο Python script ανάλυσης.....	30
Σχήμα 5.4.6.2: Ρύθμιση του περιβάλλοντος και εκτέλεση του Python script μέσω της proc_open()...	31
Σχήμα 5.4.7.1: Φόρτωση του DenseNet169 ως εξαγωγή χαρακτηριστικών και του εκπαιδευμένου ταξινομητή Naive Bayes.....	32
Σχήμα 5.4.7.2: Προεπεξεργασία της εικόνας.....	32
Σχήμα 5.4.7.3: Εξαγωγή του διανύσματος χαρακτηριστικών μέσω DenseNet169.....	33
Σχήμα 5.4.8.1: Υπολογισμός της συνημιτονικής ομοιότητας μεταξύ της εξεταζόμενης υπογραφής και των γνήσιων υπογραφών αναφοράς.....	33
Σχήμα 5.4.8.2: Συνδυασμός της πιθανότητας του ταξινομητή με τη μέση συνημιτονική ομοιότητα...	33
Σχήμα 5.4.8.3: Παραγωγή του τελικού αποτελέσματος.....	34
Σχήμα 5.4.9: Αποθήκευση του αποτελέσματος της ανάλυσης στη βάση δεδομένων μέσω της μεθόδου analyze().....	34
Σχήμα 5.4.10.1: Έλεγχος των προϋποθέσεων πριν από την εναλλακτική αξιολόγηση μέσω OpenAI.....	35
Σχήμα 5.4.10.2: Προσθήκη των εικόνων στο αίτημα.....	36
Σχήμα 5.4.10.3: Δημιουργία του prompt.....	36
Σχήμα 5.4.11.1: Δημιουργία και αποστολή του request προς το OpenAI Responses API.....	37
Σχήμα 5.4.11.2: Έλεγχος και επικύρωση του αποτελέσματος που επιστρέφει το OpenAI API.....	38
Σχήμα 5.4.11.3: Αποθήκευση των αποτελεσμάτων.....	38
Σχήμα 5.5.1.1: Ενδεικτικό παράδειγμα κοινού layout των views και εισαγωγή του δυναμικού περιεχομένου μέσω των sections του CodeIgniter.....	41
Σχήμα 5.5.1.2: Δυναμική προσαρμογή της γραμμής πλοήγησης ανάλογα με την κατάσταση σύνδεσης.....	41
Σχήμα 5.5.2.1: Πεδία τίτλου, περιγραφής και επιλογής προσώπου στη φόρμα υπογραφής.....	42
Σχήμα 5.5.2.2: Επιλογή αρχείου.....	42
Σχήμα 5.5.2.3: Δυνατότητα χαρακτηρισμού μιας νέας υπογραφής ως γνήσιας υπογραφής αναφοράς.....	43
Σχήμα 5.5.4.1: Διαχείριση πολλαπλών εικόνων και drag-and-drop κατά την προσθήκη γνήσιων υπογραφών.....	44
Σχήμα 5.5.4.2: Δημιουργία τοπικής προεπισκόπησης των επιλεγμένων εικόνων με χρήση του FileReader.....	45
Σχήμα 5.5.5.1: Υλοποίηση των φίλτρων για μέλος, κατάσταση και πηγή αποτελέσματος στη λίστα υπογραφών.....	46
Σχήμα 5.5.5.2: Δυναμική ενεργοποίηση και απενεργοποίηση φίλτρων με JavaScript ανάλογα με την επιλεγμένη κατάσταση.....	46
Σχήμα 5.5.6: Media queries για την προσαρμογή του navigation και των λειτουργιών κάμερας σε μικρότερες οθόνες.....	47

Σχήμα 6.1: Αρχική σελίδα της εφαρμογής Signalyze.	49
Σχήμα 6.2: Σελίδα σύνδεσης στη Signalyze.....	50
Σχήμα 6.3: Φόρμα δημιουργίας νέου λογαριασμού.	50
Σχήμα 6.4: Σελίδα πληροφοριών και συνοπτικής παρουσίασης των δυνατοτήτων της Signalyze.	51
Σχήμα 6.5: Σελίδα οδηγιών χρήσης και προτάσεων για την ποιότητα των εικόνων.....	52
Σχήμα 6.6: Σελίδα εισαγωγής νέας υπογραφής και ενημέρωση για την πρόοδο του συνόλου αναφοράς.	52
Σχήμα 6.7: Σελίδα μαζικής μεταφόρτωσης γνήσιων υπογραφών αναφοράς.	53
Σχήμα 6.8: Σελίδα λεπτομερειών γνήσιας υπογραφής αναφοράς.	54
Σχήμα 6.9: Σελίδα εξεταζόμενης υπογραφής πριν από την εκτέλεση των αναλύσεων.	54
Σχήμα 6.10: Σελίδα λεπτομερειών μετά την ολοκλήρωση της βασικής και της AI ανάλυσης.....	55
Σχήμα 6.11: Σελίδα συγκεντρωτικής προβολής και φιλτραρίσματος των υπογραφών.....	56
Σχήμα 6.12: Σελίδα δημιουργίας και διαχείρισης πρόσθετων μελών..	56
Σχήμα 6.13: Σελίδα προβολής των πληροφοριών και των ρυθμίσεων του προφίλ.....	57
Σχήμα 6.14: Σελίδα επεξεργασίας προφίλ, βαρών ανάλυσης και OpenAI API key..	58

Κατάλογος Πινάκων

Πίνακας 3.1: Βαρύτητες	10
Πίνακας 3.2: Αξιολογήσεις μοντέλων.....	11
Πίνακας 3.3: Αποτελέσματα αξιολόγησης βαρών	13
Πίνακας 5.1: Λειτουργικές απαιτήσεις.....	17
Πίνακας 5.2: Περιγραφή της κύριας περίπτωσης χρήσης.....	21

Συντομογραφίες

ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π. Ε.	Πτυχιακή Εργασία
TN	Τεχνητή Νοημοσύνη

Κεφάλαιο 1ο: Εισαγωγή

1.1 Ταυτοπροσωπία και φυσικές υπογραφές

Η εξακρίβωση της ταυτότητας ενός ατόμου αποτελεί θεμελιώδη διαδικασία σε ένα μεγάλο εύρος δραστηριοτήτων, όπως οι οικονομικές συναλλαγές, η πρόσβαση σε πληροφοριακά συστήματα, οι διοικητικές διαδικασίες και η υπογραφή εγγράφων. Στα πληροφοριακά συστήματα, η διαδικασία αυτή μπορεί να βασιστεί σε διαφορετικούς μηχανισμούς, μεταξύ των οποίων συγκαταλέγονται οι κωδικοί πρόσβασης, τα φυσικά μέσα ταυτοποίησης και τα βιομετρικά χαρακτηριστικά. Ως βιομετρική αναγνώριση ορίζεται γενικά η αυτοματοποιημένη αναγνώριση ενός ατόμου με βάση βιολογικά ή συμπεριφορικά χαρακτηριστικά του. Σύμφωνα με το National Institute of Standards and Technology (NIST), τα βιομετρικά στοιχεία μπορούν να χρησιμοποιηθούν είτε για την αναγνώριση της ταυτότητας ενός ατόμου είτε για την επαλήθευση μιας ήδη δηλωμένης ταυτότητας [1] [2].

Τα βιομετρικά χαρακτηριστικά διακρίνονται συνήθως σε φυσιολογικά και συμπεριφορικά. Στα φυσιολογικά χαρακτηριστικά περιλαμβάνονται στοιχεία τα οποία συνδέονται με τη φυσική δομή του ανθρώπινου σώματος, όπως τα δακτυλικά αποτυπώματα, τα χαρακτηριστικά του προσώπου και η ίριδα του ματιού. Αντίθετα, τα συμπεριφορικά βιομετρικά χαρακτηριστικά σχετίζονται με τον τρόπο με τον οποίο ένα άτομο εκτελεί μία συγκεκριμένη ενέργεια. Παραδείγματα τέτοιων χαρακτηριστικών αποτελούν ο τρόπος πληκτρολόγησης, η φωνή και η χειρόγραφη υπογραφή. Επομένως, η χειρόγραφη υπογραφή μπορεί να αντιμετωπιστεί ως ένα συμπεριφορικό βιομετρικό χαρακτηριστικό [1][3].

Παρά την ευρεία χρήση της, η χειρόγραφη υπογραφή παρουσιάζει ένα ιδιαίτερο χαρακτηριστικό το οποίο διαφοροποιεί την επαλήθευσή της από άλλες μεθόδους αυθεντικοποίησης. Δύο γνήσιες υπογραφές του ίδιου ατόμου δεν είναι συνήθως απόλυτα πανομοιότυπες. Η διαφοροποίηση αυτή είναι φυσιολογική και αναφέρεται στη σχετική βιβλιογραφία ως ενδοπροσωπική μεταβλητότητα (*intra-personal* ή *intra-class variability*). Η μορφή μιας υπογραφής είναι δυνατόν να παρουσιάζει αποκλίσεις ως προς το μέγεθος, την κλίση, την απόσταση μεταξύ επιμέρους τμημάτων και τη γενικότερη γεωμετρική της μορφή. Για τον λόγο αυτό, η σύγκριση δύο υπογραφών δεν μπορεί να βασίζεται στην απαίτηση απόλυτης οπτικής ταύτισης [3].

Στη βιβλιογραφία, τα συστήματα αυτόματης επαλήθευσης χειρόγραφων υπογραφών διακρίνονται κυρίως σε δύο κατηγορίες: στα online και στα offline συστήματα επαλήθευσης [3], [4]. Η συγκεκριμένη διάκριση δεν σχετίζεται με το εάν η εφαρμογή είναι συνδεδεμένη στο Διαδίκτυο, αλλά με τον τρόπο με τον οποίο αποκτώνται τα δεδομένα της υπογραφής.

Στα online συστήματα, η υπογραφή καταγράφεται κατά τη στιγμή της δημιουργίας της μέσω κατάλληλης συσκευής, όπως ένα tablet ή μία ψηφιακή γραφίδα. Με αυτόν τον τρόπο είναι δυνατόν να καταγραφούν δυναμικές πληροφορίες σχετικά με τη διαδικασία της υπογραφής, όπως η θέση και η κίνηση της γραφίδας, η πίεση που ασκείται και η χρονική εξέλιξη της υπογραφής [5], [6].

Αντίθετα, στα offline συστήματα η υπογραφή είναι ήδη ολοκληρωμένη και παρέχεται στο σύστημα σε μορφή στατικής εικόνας, συνήθως ύστερα από σάρωση ή φωτογράφιση ενός εγγράφου. Σε αυτή την περίπτωση δεν είναι διαθέσιμες οι δυναμικές πληροφορίες που αφορούν τον τρόπο με τον οποίο δημιουργήθηκε η υπογραφή. Το σύστημα διαθέτει μόνο τα οπτικά χαρακτηριστικά της τελικής εικόνας και καλείται να εξαγάγει από αυτά τις απαραίτητες πληροφορίες για την επαλήθευσή της. Η απουσία των δυναμικών αυτών πληροφοριών καθιστά την offline επαλήθευση ιδιαίτερα απαιτητικό πρόβλημα [3], [4].

Η παρούσα εργασία εντάσσεται στην κατηγορία της offline επαλήθευσης χειρόγραφων υπογραφών, καθώς οι υπογραφές παρέχονται στο σύστημα ως ψηφιακές εικόνες. Η ανάπτυξη μεθόδων που μπορούν να πραγματοποιούν αξιόπιστα τη διάκριση μεταξύ γνήσιων και πλαστών υπογραφών αποτελεί

σημαντικό ερευνητικό πεδίο, ιδιαίτερα εξαιτίας της φυσικής μεταβλητότητας των γνήσιων υπογραφών και της δυνατότητας δημιουργίας πλαστών υπογραφών με υψηλό βαθμό ομοιότητας [3], [4].

1.2 Το πρόβλημα της πλαστογράφησης υπογραφών

Λόγω της ευρείας χρήσης των χειρόγραφων υπογραφών ως μέσου ταυτοποίησης των προσώπων και ως έκφρασης της συγκατάθεσής τους, το ζήτημα της πλαστογράφησης αποκτά ιδιαίτερη σημασία. Σήμερα, οι χειρόγραφες υπογραφές εξακολουθούν να χρησιμοποιούνται σε τραπεζικές διαδικασίες, πληρωμές και οικονομικές συναλλαγές, καθώς και σε επιταγές, δανειακές συμβάσεις, συμβάσεις ακινήτων και άλλα οικονομικά και νομικά έγγραφα [4], [7]. Επομένως, η αξιόπιστη επαλήθευση της γνησιότητας των υπογραφών αποτελεί αναγκαία προϋπόθεση για την ασφαλή διεξαγωγή των σχετικών δραστηριοτήτων.

Η παραποίηση μιας υπογραφής μπορεί να έχει σοβαρές οικονομικές και νομικές συνέπειες. Η χρήση μιας παραποιημένης υπογραφής μπορεί να δημιουργήσει ψευδή ένδειξη έγκρισης μιας οικονομικής συναλλαγής, αποδοχής των όρων μιας συμφωνίας ή συγκατάθεσης στο περιεχόμενο ενός εγγράφου. Με αυτόν τον τρόπο, η παραποίηση υπογραφών μπορεί να προκαλέσει οικονομικές απώλειες, απάτη ή αμφισβήτηση της αυθεντικότητας οικονομικών και νομικών εγγράφων [7].

Την ίδια στιγμή η παραποίηση μιας υπογραφής μπορεί επίσης να συνδεθεί με την παραποίηση της ταυτότητας ενός προσώπου. Όπως αναφέρθηκε παραπάνω οι χειρόγραφες υπογραφές χρησιμοποιούνται συχνά ως μέσο ταυτοποίησης ή επιβεβαίωσης μιας πράξης ενός προσώπου και η επιτυχής παραποίηση τέτοιων υπογραφών μπορεί να οδηγήσει σε παραπλανητικές πληροφορίες για την πραγματική εκτέλεση της πράξης από αυτό το πρόσωπο [4],[7].

Η επαλήθευση της γνησιότητας μιας υπογραφής δεν είναι πάντοτε εύκολη. Όπως αναφέρθηκε παραπάνω, διαφορετικές γνήσιες υπογραφές του ίδιου προσώπου παρουσιάζουν φυσική μεταβλητότητα. Παράλληλα, μια πλαστή υπογραφή που έχει δημιουργηθεί με συνειδητή μίμηση γνήσιων δειγμάτων μπορεί να μοιάζει σε μεγάλο βαθμό με την αυθεντική [3], [8]. Η συνδυασμένη επίδραση αυτών των δύο παραγόντων καθιστά την επαλήθευση υπογραφών ιδιαίτερα απαιτητική.

Η παραδοσιακή επαλήθευση υπογραφών μέσω οπτικής σύγκρισης παρουσιάζει σημαντικές πρακτικές δυσκολίες, καθώς η αξιολόγηση επηρεάζεται τόσο από τη φυσική μεταβλητότητα των γνήσιων υπογραφών όσο και από την ομοιότητα ορισμένων πλαστών δειγμάτων με τα αυθεντικά. Για την αντιμετώπιση αυτών των περιορισμών, η αυτοματοποίηση της επαλήθευσης υπογραφών αποτελεί αντικείμενο επιστημονικής έρευνας εδώ και δεκαετίες, με στόχο την ανάπτυξη συστημάτων που μπορούν να διακρίνουν αξιόπιστα τις γνήσιες από τις πλαστές υπογραφές [8].

Στην περίπτωση της offline επαλήθευσης, η δυσκολία εντείνεται επειδή το σύστημα διαθέτει μόνο την τελική εικόνα της υπογραφής και όχι πληροφορίες για τη διαδικασία δημιουργίας της, όπως η πίεση του στυλό, η ταχύτητα και η χρονική σειρά των κινήσεων. Κατά συνέπεια, το σύστημα ή ο επαληθευτής καλείται να στηρίξει την απόφασή του αποκλειστικά στα οπτικά χαρακτηριστικά της τελικής εικόνας [3], [4].

1.3 Ανίχνευση πλαστών υπογραφών μέσω TN

Η Τεχνητή Νοημοσύνη μπορεί να συμβάλει σημαντικά στην αυτοματοποίηση της διαδικασίας επαλήθευσης χειρόγραφων υπογραφών. Μέσω της ανάλυσης μεγάλου αριθμού γνήσιων και πλαστών δειγμάτων, ένα σύστημα μπορεί να μάθει να εντοπίζει πρότυπα και διαφοροποιήσεις που σχετίζονται με τη γνησιότητα μιας υπογραφής [4], [9].

Σε αντίθεση με την απλή οπτική σύγκριση, οι μέθοδοι Τεχνητής Νοημοσύνης μπορούν να αξιοποιήσουν ένα σύνολο χαρακτηριστικών της υπογραφής και να εκτιμήσουν κατά πόσο ένα νέο δείγμα παρουσιάζει ομοιότητες με γνωστές γνήσιες υπογραφές ή χαρακτηριστικά που συναντώνται σε πλαστογραφήσεις [3], [4]. Η δυνατότητα αυτή είναι ιδιαίτερα σημαντική εξαιτίας της φυσικής μεταβλητότητας των γνήσιων υπογραφών και της οπτικής ομοιότητας που μπορεί να παρουσιάζουν ορισμένες επιδέξιες πλαστογραφήσεις.

Η εφαρμογή αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης έχει καταστήσει δυνατή τη δημιουργία συστημάτων τα οποία μπορούν να αναλύουν και να αξιολογούν υπογραφές. Συγκεκριμένα οι πιο νέοι αλγόριθμοι βαθιάς μάθησης έχουν σημειώσει μεγάλη πρόοδο καθώς μπορούν να ανιχνεύουν και να μαθαίνουν πολύτιμα χαρακτηριστικά απευθείας από τις εικόνες των υπογραφών [4],[9].

Έτσι η εφαρμογή της Τεχνητής Νοημοσύνης μπορεί να προσφέρει πολύ γρηγορότερη, αυτοματοποιημένη και οργανωμένη διαδικασία επαλήθευσης ιδιαίτερα σε περιπτώσεις που πρέπει να αξιολογηθούν πολλές υπογραφές. Παρ' όλα αυτά η απόδοσή της εξαρτάται από μεταβλητές όπως η διαθεσιμότητα δεδομένων και οι εγγενείς διαφορές των υπογραφών και η δυσκολία της παραποίησης [3],[4].

Αν και η Τεχνητή Νοημοσύνη δεν μπορεί να λύσει τα προβλήματα που συνοδεύουν την αξιολόγηση των χειρόγραφων υπογραφών αποτελεί ένα πολύ σημαντικό εργαλείο για τη δημιουργία αυτοματοποιημένων συστημάτων για την καταπολέμηση της παραποίησης. Αυτό είναι και ο λόγος που η χρήση αλγορίθμων μηχανικής μάθησης και βαθιάς μάθησης στην εκτός σύνδεσης επαλήθευση υπογραφών αποτελεί ένα σχετικά νέο πεδίο έρευνας [4],[9].

1.4 Κίνητρο

Το κίνητρο για την εκπόνηση της παρούσας εργασίας προέκυψε από τα αποτελέσματα προηγούμενης πτυχιακής εργασίας με αντικείμενο την ανίχνευση πλαστογραφίας υπογραφών μέσω τεχνικών Μηχανικής και Βαθιάς Μάθησης [10]. Στην εργασία αυτή μελετήθηκαν και αξιολογήθηκαν διαφορετικά μοντέλα και συνδυασμοί τεχνικών, αναδεικνύοντας τις δυνατότητες της Τεχνητής Νοημοσύνης στην αποτελεσματική διάκριση μεταξύ γνήσιων και πλαστών υπογραφών.

Ωστόσο, η προηγούμενη μελέτη επικεντρώθηκε κυρίως στην ανάπτυξη και την πειραματική αξιολόγηση των μοντέλων, χωρίς να παρέχει μια ολοκληρωμένη και εύχρηστη εφαρμογή μέσω της οποίας ένας απλός χρήστης θα μπορούσε να αξιοποιήσει τα αποτελέσματα αυτά στην πράξη. Η χρήση ενός μοντέλου ανίχνευσης απαιτούσε επομένως τεχνικές γνώσεις και δεν υπήρχε ένας απλός τρόπος με τον οποίο ο ενδιαφερόμενος θα μπορούσε να εισαγάγει δικά του δείγματα υπογραφών και να πραγματοποιήσει έναν έλεγχο γνησιότητας.

Το γεγονός αυτό αποτέλεσε το βασικό κίνητρο της παρούσας πτυχιακής εργασίας. Στόχος είναι η αξιοποίηση των δυνατοτήτων που προσφέρουν οι τεχνικές Τεχνητής Νοημοσύνης και η ενσωμάτωσή τους σε ένα ολοκληρωμένο διαδικτυακό περιβάλλον, το οποίο καθιστά τη διαδικασία ελέγχου μιας υπογραφής περισσότερο προσιτή και εύχρηστη. Με αυτόν τον τρόπο επιχειρείται η μετάβαση από την πειραματική μελέτη των μοντέλων σε μία πρακτική εφαρμογή, μέσω της οποίας ο χρήστης μπορεί να αξιοποιήσει τις δυνατότητες αυτόματης επαλήθευσης υπογραφών χωρίς να απαιτούνται εξειδικευμένες γνώσεις Μηχανικής Μάθησης.

1.5 Συνεισφορά

Η κύρια συνεισφορά της παρούσας πτυχιακής εργασίας είναι η ανάπτυξη της διαδικτυακής εφαρμογής Signalyze, η οποία υποστηρίζει την ανίχνευση πιθανών πλαστών χειρόγραφων υπογραφών μέσα από

ένα εύχρηστο περιβάλλον. Η εφαρμογή αξιοποιεί μεθόδους Τεχνητής Νοημοσύνης και επιτρέπει στον χρήστη να καταχωρίζει γνήσια δείγματα αναφοράς και, στη συνέχεια, να αξιολογεί νέες υπογραφές.

Αρχικά ο χρήστης πρέπει να δημιουργήσει έναν λογαριασμό και να συνδεθεί στην εφαρμογή. Χρειάζονται τουλάχιστον πέντε γνήσιες υπογραφές για να ξεκινήσει η διαδικασία επαλήθευσης. Αυτές θα λειτουργήσουν ως αναφορές κατά την αξιολόγηση. Μπορούν να ανεβαστούν είτε ως ψηφιακές εικόνες είτε να καταγραφούν από την κάμερα μιας φορητής συσκευής.

Μόλις ανεβαστούν οι επαρκείς υπογραφές ο χρήστης μπορεί να υποβάλει μια νέα για να αξιολογηθεί. Η Signalyze παρέχει στον χρήστη δύο μεθόδους αξιολόγησης. Η πρώτη είναι η χρήση της συνδυασμένης βαθμολογίας αυθεντικότητας η οποία βασίζεται στο αποτέλεσμα ενός εκπαιδευμένου μοντέλου και της ομοιότητας της υπογραφής που αξιολογείται με τις γνήσιες που έχουν ανεβεί. Είναι δυνατόν ο χρήστης να καθορίσει πόσο σημαντική είναι η βαρύτητα κάθε παράγοντα για το αποτέλεσμα.

Η δεύτερη μέθοδος χρησιμοποιεί την υπηρεσία Τεχνητής Νοημοσύνης OpenAI για να παρέχει μια εναλλακτική αξιολόγηση της υπογραφής. Με αυτόν τον τρόπο η εφαρμογή δίνει στον χρήστη την ευκαιρία να πραγματοποιήσει δύο αξιολογήσεις και να τις συγκρίνει.

Επιπρόσθετα, η Signalyze επιτρέπει στον χρήστη να προσθέσει έως πέντε πρόσθετα μέλη στον λογαριασμό του. Για κάθε μέλος μπορεί να αποθηκευτεί ανεξάρτητο σύνολο γνήσιων υπογραφών και να πραγματοποιούνται ξεχωριστοί έλεγχοι, χωρίς να επηρεάζονται τα δεδομένα του κύριου χρήστη ή των άλλων μελών. Επομένως, ο ίδιος λογαριασμός μπορεί να χρησιμοποιείται για την αξιολόγηση υπογραφών περισσότερων προσώπων.

Συνοψίζοντας η προσφορά αυτού του έργου δεν καλύπτει μόνο την υλοποίηση των υπάρχοντων μοντέλων ανίχνευσης πλαστογραφίας αλλά τα μετατρέπει σε μια λειτουργική εφαρμογή με μια βολική διεπαφή.

1.6 Οργάνωση της εργασίας

Αυτή η εργασία αποτελείται από επτά μεγάλα κεφάλαια που καλύπτουν από την προσέγγιση του προβλήματος και την επίλυσή του μέχρι την υλοποίηση και την αξιολόγηση της εφαρμογής Signalyze.

Το Κεφάλαιο 1 θέτει το γενικό πλαίσιο της εργασίας. Αρχικά παρουσιάζει την έννοια της ταυτοπροσωπίας και τον ρόλο των χειρόγραφων υπογραφών ως μέσου ταυτοποίησης και επιβεβαίωσης πράξεων. Στη συνέχεια εξετάζει το πρόβλημα της πλαστογράφησης χειρόγραφων υπογραφών και τις δυνατότητες των τεχνικών Τεχνητής Νοημοσύνης για την αντιμετώπισή του. Τέλος, παρουσιάζει το κίνητρο και την κύρια συνεισφορά της εργασίας.

Το Κεφάλαιο 2 δίνει μια ανασκόπηση της σχετικής βιβλιογραφίας. Αυτό το μέρος αποτελείται από δύο μέρη: την έρευνα σε μοντέλα Τεχνητής Νοημοσύνης για την ανίχνευση παραποίησης και σε συστήματα που ενσωματώνουν αυτές τις τεχνικές.

Το Κεφάλαιο 3 παρέχει πληροφορίες για την πειραματική μελέτη που έγινε για να επιλεγεί η μέθοδος ανάλυσης που ενσωματώνεται στην Signalyze. Περιέχει την περιγραφή του συνόλου δεδομένων, την πειραματική διάταξη, τον συνδυασμό των μοντέλων που εξετάστηκαν και τα αποτελέσματα που βοήθησαν στην επιλογή του DenseNet169 σε συνδυασμό με το Naive Bayes ως τον καλύτερο συνδυασμό. Επίσης, συζητά το θέμα των διαφόρων συντελεστών βάρους μεταξύ της πιθανότητας της ταξινόμησης και της συνημιτονικής ομοιότητας για να προσδιοριστούν οι προεπιλεγμένες παράμετροι της εφαρμογής.

Το Κεφάλαιο 4 περιέχει πληροφορίες για τις γλώσσες προγραμματισμού, τα πλαίσια και τις τεχνολογίες που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής. Ειδικότερα, παρουσιάζονται το CodeIgniter 4, η PHP, η MySQL, η JavaScript και το Bootstrap, καθώς και βιβλιοθήκες και εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση της μηχανικής μάθησης και των υποσυστημάτων επεξεργασίας εικόνας.

Το Κεφάλαιο 5 συζητά τον σχεδιασμό και την υλοποίηση της Signalyze. Ξεκινά με τις λειτουργικές απαιτήσεις και τις κύριες περιπτώσεις χρήσης. Στη συνέχεια δίνει λεπτομέρειες για τη δομή της βάσης δεδομένων, την υλοποίηση του back-end και του front-end της εφαρμογής. Δίνεται μεγάλη προσοχή σε σημαντικά θέματα όπως η αποστολή και η διαχείριση υπογραφών, η επικοινωνία μεταξύ PHP και Python scripts, η διαδικασία ανάλυσης και η διαχείριση των αποτελεσμάτων.

Το Κεφάλαιο 6 παρουσιάζει την τελική έκδοση της εφαρμογής μέσα από τις κύριες οθόνες της. Περιγράφει πώς ο χρήστης κινείται από την εγγραφή και τη σύνδεση στη δημιουργία του συνόλου αναφοράς, την υποβολή μιας νέας υπογραφής, την ανάλυση και την προβολή των αποτελεσμάτων. Επίσης, περιγράφει τη λειτουργικότητα της διαχείρισης των μελών και των προφίλ.

Τέλος, το Κεφάλαιο 7 συζητά τα κύρια συμπεράσματα αυτής της εργασίας και προτείνει κάποιες πιθανές κατευθύνσεις για τη μελλοντική ανάπτυξη της Signalyze.

Κεφάλαιο 2ο: Σχετικές εργασίες

Στο παρόν κεφάλαιο παρουσιάζονται συνοπτικά ερευνητικές εργασίες που σχετίζονται με την αυτόματη επαλήθευση και την ανίχνευση πλαστών χειρόγραφων υπογραφών. Στην Ενότητα 2.1 εξετάζονται εργασίες που επικεντρώνονται στην ανάπτυξη και αξιολόγηση μοντέλων Τεχνητής Νοημοσύνης για τη διάκριση μεταξύ γνήσιων και πλαστών υπογραφών. Στη συνέχεια, στην Ενότητα 2.2 παρουσιάζονται εργασίες στις οποίες αντίστοιχες τεχνικές έχουν ενσωματωθεί σε ολοκληρωμένες εφαρμογές ή πρωτότυπα συστήματα, με στόχο την πρακτική αξιοποίησή τους από τελικούς χρήστες.

2.1 Μοντέλα TN για ανίχνευση πλαστών υπογραφών

Σε αυτό το πλαίσιο ο Χαφέμαν, η Σαμπορίν και η Ολιβείρα πρότειναν τη χρήση μιας τεχνικής βαθιάς μάθησης για την εκτός σύνδεσης επαλήθευση υπογραφών με στόχο την αυτόματη εξαγωγή χαρακτηριστικών απευθείας από εικόνες. Συγκεκριμένα στην έρευνά τους χρησιμοποίησαν ένα βαθύ νευρωνικό δίκτυο συσσωρευτικής συσχέτισης για την εξαγωγή των χαρακτηριστικών που χρησιμοποιούνται για την ταξινόμηση γνήσιων και πλαστών υπογραφών. Έθεσαν σε δοκιμή τον αλγόριθμό τους σε διάφορα σύνολα δεδομένων όπως το GPDS-960 και απέδειξαν ότι τα χαρακτηριστικά που έμαθε το βαθύ δίκτυο μπορούν να εφαρμοστούν επιτυχώς ακόμη και σε χρήστες που δεν έχουν εκπαιδευτεί [11].

Από την άλλη ο Λιου και συνεργάτες του παρουσίασαν μια άλλη τεχνική βασισμένη στην ιδέα της βαθιάς μετρικής μάθησης και επικεντρώθηκε στην σύγκριση των υπο-περιοχών των υπογραφών. Η τεχνική που ονομάζεται Mutual Signature DenseNet έχει ως στόχο να συγκρίνει μικρές και τοπικές διαφορές μεταξύ δύο υπογραφών αντί να βασίζεται μόνο στην παγκόσμια μία. Οι βαθμοί ομοιότητας διαφόρων περιοχών συνδυάζονται για να ληφθεί η απόφαση. Ο αλγόριθμος δοκιμάστηκε στα σύνολα δεδομένων CEDAR και GPDS και έδειξε ιδιαίτερα υποσχόμενη απόδοση τόσο σε περιπτώσεις που εξαρτώνται από τον συγγραφέα όσο και σε περιπτώσεις που δεν εξαρτώνται από αυτόν [12].

Η δυνατότητα μεταφοράς γνώσης από το πρόβλημα της αναγνώρισης χειρόγραφου κειμένου στο πρόβλημα της επαλήθευσης υπογραφών διερευνήθηκε από τον Τουρούνις και συνεργάτες του. Η ιδέα αυτής της προσέγγισης περιλαμβάνει τη χρήση του προηγούμενως εκπαιδευμένου μοντέλου σχεδιασμένου για την αναγνώριση χειρόγραφου κειμένου και την προσαρμογή του για την εκτός σύνδεσης επαλήθευση υπογραφών. Με αυτόν τον τρόπο επιδιώκεται να μειωθεί η απαίτηση για μεγάλο αριθμό δειγμάτων υπογραφών για την εκπαίδευση [13].

Τέλος, οι Xiao και Wu παρουσίασαν μια τεχνική offline επαλήθευσης υπογραφών με χρήση Spatial Transformer Network. Στόχος της τεχνικής ήταν να επιτρέψει στο μοντέλο να εντοπίζει και να εστιάζει στις σημαντικές περιοχές της εικόνας της υπογραφής, ώστε να διαχειρίζεται αποτελεσματικότερα διαφορές στη θέση, στον προσανατολισμό και στο σχήμα των επιμέρους στοιχείων. Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη στην περίπτωση επιδέξιων πλαστογραφήσεων, όπου οι διαφορές μεταξύ γνήσιων και πλαστών υπογραφών μπορεί να είναι περιορισμένες [14].

2.2 Εφαρμογές ανίχνευσης και επαλήθευσης υπογραφών

Εκτός από την ανάπτυξη και αξιολόγηση μοντέλων, έχουν πραγματοποιηθεί ερευνητικές προσπάθειες για την ενσωμάτωση τεχνικών επαλήθευσης υπογραφών σε εφαρμογές που μπορούν να χρησιμοποιηθούν σε περισσότερο πρακτικά σενάρια. Οι εργασίες αυτές παρουσιάζουν ιδιαίτερο ενδιαφέρον για την παρούσα πτυχιακή, καθώς προσεγγίζουν το πρόβλημα όχι μόνο από την πλευρά της

ακρίβειας ενός αλγορίθμου, αλλά και από την πλευρά της αλληλεπίδρασης του τελικού χρήστη με ένα ολοκληρωμένο σύστημα.

Οι **Fatihia, Fariza και Karlita** ανέπτυξαν το **VeriSign**, μία εφαρμογή για Android συσκευές με σκοπό την offline επαλήθευση χειρόγραφων υπογραφών. Η εφαρμογή ενσωματώνει ένα συνελκτικό νευρωνικό δίκτυο και επιτρέπει την αξιολόγηση εικόνων υπογραφών απευθείας σε κινητή συσκευή. Η προτεινόμενη μέθοδος πέτυχε ακρίβεια 86,36%, ενώ οι συγγραφείς αξιολόγησαν επίσης δείκτες όπως FAR, FRR και EER. Ένα σημαντικό στοιχείο της συγκεκριμένης εργασίας είναι η προσπάθεια μεταφοράς ενός μοντέλου από το καθαρά πειραματικό περιβάλλον σε μία πραγματική mobile εφαρμογή, με στόχο να είναι δυνατή η πρακτική χρήση του από smartphone [15].

Οι **Alonso-Fernandez et al.** ανέπτυξαν ένα πρωτότυπο **web-based** σύστημα ασφαλούς πρόσβασης μέσω επαλήθευσης υπογραφής [16]. Η αρχιτεκτονική του συστήματος περιλάμβανε ένα Tablet PC για την εισαγωγή της υπογραφής από τον χρήστη, έναν web server και έναν ξεχωριστό server για την επαλήθευση των υπογραφών. Ο χρήστης παρείχε την υπογραφή του μέσω του Tablet PC και τα δεδομένα μεταφέρονταν μέσω δικτύου για την πραγματοποίηση της επαλήθευσης. Ιδιαίτερη έμφαση δόθηκε στην επεκτασιμότητα της αρχιτεκτονικής, ώστε να μπορούν να χρησιμοποιηθούν διαφορετικοί αισθητήρες και μεγαλύτερες βάσεις δεδομένων, καθώς και σε ζητήματα ασφάλειας και ιδιωτικότητας. Αν και πρόκειται για online και όχι offline επαλήθευση, η εργασία παρουσιάζει μία από τις χαρακτηριστικές περιπτώσεις όπου η τεχνολογία signature verification ενσωματώνεται σε ολοκληρωμένη web αρχιτεκτονική [16].

Οι **Roszczewska και Niewiadomska-Szynkiewicz** μελέτησαν την αξιοποίηση κινητών συσκευών για τη βιομετρική επαλήθευση μέσω χειρόγραφης υπογραφής. Στην εργασία τους αναπτύχθηκε ειδική εφαρμογή για τη συλλογή υπογραφών από smartphones, μέσω της οποίας καταγράφονταν στοιχεία όπως οι συντεταγμένες της κίνησης και η πίεση κατά τη διαδικασία υπογραφής. Τα δεδομένα χρησιμοποιήθηκαν για την αξιολόγηση διαφορετικών μοντέλων νευρωνικών δικτύων. Η μελέτη πραγματοποιήθηκε σε δείγματα 53 ατόμων και παρουσίασε ιδιαίτερα χαμηλό Equal Error Rate στις τυχαίες πλαστογραφήσεις και υψηλότερο, αλλά επίσης αξιοσημείωτο, αποτέλεσμα στις επιδέξιες πλαστογραφήσεις. Η συγκεκριμένη εργασία δείχνει πως ένα σύστημα επαλήθευσης μπορεί να ενσωματωθεί φυσικά στη χρήση μιας καθημερινής φορητής συσκευής [6].

Μια ακόμη εφαρμοσμένη προσέγγιση παρουσιάστηκε στο **iVerify**, ένα σύστημα offline αναγνώρισης και επαλήθευσης υπογραφών για Android συσκευές. Το σύστημα σχεδιάστηκε ώστε να μπορεί να σαρώσει μία χειρόγραφη υπογραφή από έγγραφο, να την αναγνωρίσει και στη συνέχεια να τη χρησιμοποιήσει ως βάση για την επαλήθευση άλλων υπογραφών. Η επεξεργασία βασίστηκε σε τεχνικές τεχνητών νευρωνικών δικτύων και η εφαρμογή αξιολογήθηκε ως προς τη λειτουργικότητά της μέσω δοκιμών Boundary Value Analysis. Παρότι πρόκειται περισσότερο για εφαρμοσμένο/capstone έργο και όχι για εργασία αντίστοιχου ερευνητικού βάθους με τις προηγούμενες, παρουσιάζει ενδιαφέρον ως παράδειγμα μεταφοράς της offline signature verification σε ένα απλό περιβάλλον Android [17].

Οι παραπάνω εφαρμογές παρουσιάζουν διαφορετικές προσεγγίσεις στη μεταφορά των συστημάτων επαλήθευσης υπογραφών από το ερευνητικό επίπεδο στο επίπεδο της πραγματικής χρήσης. Παρατηρούνται λύσεις για κινητές συσκευές, συστήματα που αξιοποιούν δεδομένα τα οποία συλλέγονται κατά τη στιγμή της υπογραφής και web-based αρχιτεκτονικές για απομακρυσμένη επαλήθευση. Η **Signalize** διαφοροποιείται κυρίως στο ότι αποτελεί web εφαρμογή offline επαλήθευσης, βασίζεται σε πολλαπλές γνήσιες υπογραφές αναφοράς ανά χρήστη και παρέχει στον τελικό χρήστη δύο εναλλακτικούς μηχανισμούς αξιολόγησης.

Κεφάλαιο 3ο: Πειραματική μελέτη

Στο παρόν κεφάλαιο παρουσιάζεται η πειραματική μελέτη που πραγματοποιήθηκε με στόχο την επιλογή της καταλληλότερης μεθόδου για την αξιολόγηση της γνησιότητας των χειρόγραφων υπογραφών στην εφαρμογή **Signalyze**. Αρχικά παρουσιάζεται ο σκοπός της πειραματικής διαδικασίας και το σύνολο δεδομένων που χρησιμοποιήθηκε. Στη συνέχεια περιγράφεται η πειραματική εγκατάσταση, η οποία περιλαμβάνει δύο διαδοχικές φάσεις: τη συγκριτική αξιολόγηση διαφορετικών συνδυασμών μοντέλων βαθιάς και μηχανικής μάθησης και τη διερεύνηση διαφορετικών βαρών κατά τον συνδυασμό της πρόβλεψης του επιλεγμένου μοντέλου με τον βαθμό ομοιότητας της εξεταζόμενης υπογραφής προς γνωστά γνήσια δείγματα. Τέλος, παρουσιάζονται και αναλύονται τα αποτελέσματα των πειραμάτων, βάσει των οποίων πραγματοποιήθηκε η τελική επιλογή του μοντέλου και της προεπιλεγμένης βαρύτητας που χρησιμοποιείται από την εφαρμογή.

3.1 Σκοπός πειραματικής μελέτης

Ο κύριος στόχος της πειραματικής μελέτης ήταν να εντοπιστούν διάφοροι τρόποι ταξινόμησης των αυθεντικών υπογραφών και των πλαστών και να επιλεγεί η βέλτιστη προσέγγιση που θα εφαρμοστεί στο **Signalyze**. Για το λόγο αυτό πραγματοποιήθηκαν δύο διαδοχικά σύνολα πειραμάτων.

Στην πρώτη φάση αξιολογήθηκαν διάφοροι συνδυασμοί τεχνικών βαθιάς μάθησης για την εξαγωγή χαρακτηριστικών από εικόνες υπογραφών και αλγορίθμων μηχανικής μάθησης για την τελική ταξινόμησή τους. Όλοι οι συνδυασμοί δοκιμάστηκαν στο ίδιο σύνολο δεδομένων και υπό παρόμοιες συνθήκες, ώστε να είναι δυνατή η σύγκριση της απόδοσής τους και η επιλογή του καλύτερου.

Στη δεύτερη φάση, το μοντέλο με την καλύτερη απόδοση στην πρώτη δοκιμή χρησιμοποιήθηκε ως βάση για περαιτέρω αξιολόγηση. Η λειτουργία της εφαρμογής δεν εξαρτάται μόνο από την πιθανότητα που υπολογίζει το μοντέλο, αλλά από τον συνδυασμό αυτής της εκτίμησης με τον βαθμό ομοιότητας της εξεταζόμενης υπογραφής προς τα γνήσια δείγματα. Για τον λόγο αυτό, δοκιμάστηκαν διαφορετικοί συνδυασμοί βαρύτητας των δύο εκτιμήσεων, ώστε να εντοπιστεί ο αποτελεσματικότερος.

Αυτός ο συγκεκριμένος συνδυασμός είναι κρίσιμος για την πειραματική μελέτη γιατί το αποτέλεσμα του επιλέγεται ως προεπιλεγμένη ρύθμιση για το **Signalyze**. Επιπλέον θα πρέπει να σημειωθεί ότι η εφαρμογή επιτρέπει στον χρήστη να αλλάξει την ισορροπία μεταξύ αυτών των δύο παραμέτρων σύμφωνα με τις προτιμήσεις του.

Έτσι οι κύριοι στόχοι της πειραματικής μελέτης ήταν να επιλεγεί το βέλτιστο μοντέλο ταξινόμησης και να προσδιοριστεί η προεπιλεγμένη αναλογία βάρους μεταξύ της εκτίμησης του μοντέλου και του επιπέδου ομοιότητας.

3.2 Σύνολο δεδομένων

Για την πραγματοποίηση της πειραματικής μελέτης χρησιμοποιήθηκε μέρος του δημόσια διαθέσιμου συνόλου δεδομένων **Signature Verification Dataset**, το οποίο διατίθεται μέσω της πλατφόρμας **Kaggle** [18]. Το συγκεκριμένο σύνολο δεδομένων περιλαμβάνει εικόνες γνήσιων και πλαστών χειρόγραφων υπογραφών, οργανωμένες σε διαφορετικούς φακέλους ανά υπογράφοντα.

Στην παρούσα εργασία χρησιμοποιήθηκαν τα δεδομένα που αντιστοιχούν στους υπογράφοντες με αναγνωριστικά από **049 έως και 069**. Για κάθε υπογράφοντα υπάρχει ένας φάκελος που περιλαμβάνει γνήσιες υπογραφές και ένας αντίστοιχος φάκελος με την κατάληξη `_forg`, ο οποίος περιλαμβάνει

πλαστές υπογραφές. Για παράδειγμα, ο φάκελος 049 περιλαμβάνει γνήσια δείγματα του συγκεκριμένου ατόμου, ενώ ο φάκελος 049_forg περιλαμβάνει τις αντίστοιχες πλαστές υπογραφές.

Συνολικά χρησιμοποιήθηκαν δεδομένα από **21 διαφορετικούς υπογράφοντες**. Για κάθε άτομο επιλέχθηκαν πέντε γνήσιες και πέντε πλαστές υπογραφές, με αποτέλεσμα το σύνολο αξιολόγησης να αποτελείται από **210 εικόνες**, εκ των οποίων οι 105 ανήκαν στην κατηγορία των γνήσιων και οι 105 στην κατηγορία των πλαστών υπογραφών. Η ισόποση παρουσία των δύο κατηγοριών δημιουργεί ένα ισορροπημένο σύνολο αξιολόγησης και διευκολύνει τη συγκριτική μελέτη των διαφορετικών προσεγγίσεων.

Οι συγκεκριμένες εικόνες χρησιμοποιήθηκαν ως δεδομένα αξιολόγησης των ήδη εκπαιδευμένων μοντέλων και για τη διερεύνηση των διαφορετικών συνδυασμών βαρύτητας που περιγράφονται στις επόμενες ενότητες.

3.3 Πειραματική εγκατάσταση

Η πειραματική διαδικασία χωρίστηκε σε δύο βασικές φάσεις. Στην πρώτη πραγματοποιήθηκε η ανεξάρτητη αξιολόγηση των διαθέσιμων μοντέλων, ενώ στη δεύτερη εξετάστηκε η απόδοση του συνδυαστικού μηχανισμού αξιολόγησης που χρησιμοποιεί την πρόβλεψη του επιλεγμένου μοντέλου και την ομοιότητα προς γνήσια δείγματα.

Για την πρώτη φάση χρησιμοποιήθηκε το Python script `benchmark_models.py`, το οποίο φορτώνει τα ήδη εκπαιδευμένα μοντέλα χωρίς να πραγματοποιεί νέα εκπαίδευση ή τροποποίησή τους. Για κάθε έναν από τους 21 υπογράφοντες επιλέγονται πέντε γνήσιες και πέντε πλαστές εικόνες, οι οποίες χρησιμοποιούνται για την κοινή αξιολόγηση όλων των μοντέλων.

Πριν από την εξαγωγή των χαρακτηριστικών, όλες οι εικόνες μετατρέπονται σε RGB και προσαρμόζονται σε διαστάσεις **224 × 224 pixels**.

Στην αξιολόγηση χρησιμοποιήθηκαν οκτώ διαφορετικές αρχιτεκτονικές βαθιάς μάθησης:

- DenseNet121
- DenseNet169
- DenseNet201
- EfficientNetV2B0
- MobileNet
- MobileNetV2
- VGG16
- VGG19

Οι παραπάνω αρχιτεκτονικές χρησιμοποιήθηκαν ως εξαγωγείς χαρακτηριστικών και φορτώθηκαν με προεκπαιδευμένα βάρη του ImageNet, χωρίς το τελικό επίπεδο ταξινόμησης. Τα χαρακτηριστικά που παράγονταν από κάθε εικόνα μετατρέπονταν σε μονοδιάστατο διάνυσμα και τροφοδοτούνταν στον αντίστοιχο εκπαιδευμένο ταξινομητή μηχανικής μάθησης.

Για κάθε αρχιτεκτονική εξετάστηκαν τέσσερις ταξινομητές:

- Logistic Regression
- Decision Tree

- Random Forest
- Naive Bayes

Με αυτόν τον τρόπο αξιολογήθηκαν συνολικά **32 διαφορετικοί συνδυασμοί**.

Για κάθε υπογραφή, ο ταξινομητής παρήγαγε την πιθανότητα να ανήκει στη γνήσια κατηγορία. Ως όριο απόφασης χρησιμοποιήθηκε η τιμή **0,50**, επομένως τιμή μεγαλύτερη ή ίση του 0,50 οδηγούσε σε χαρακτηρισμό της υπογραφής ως γνήσιας, ενώ μικρότερη τιμή σε χαρακτηρισμό της ως πλαστής.

Η απόδοση αξιολογήθηκε με βάση τις μετρικές **Ακρίβεια ταξινόμησης, Θετική προγνωστική αξία, Ανάκληση και F1-score**, καθώς και τις τιμές **Αληθώς θετικά (TP), Αληθώς αρνητικά (TN), Ψευδώς θετικά (FP) και Ψευδώς αρνητικά (FN)**.

Με βάση τα αποτελέσματα της πρώτης φάσης, επιλέχθηκε ο συνδυασμός **DenseNet169 – Naive Bayes**, ο οποίος παρουσίασε την υψηλότερη συνολική ακρίβεια μεταξύ των μοντέλων που εξετάστηκαν.

Για τη δεύτερη φάση χρησιμοποιήθηκε το script `benchmark_signature_detection_v3.py`. Σε αυτή την περίπτωση το αποτέλεσμα δεν προκύπτει αποκλειστικά από την πιθανότητα που παρέχει ο ταξινομητής. Για κάθε εξεταζόμενη υπογραφή υπολογίζονται δύο επιμέρους τιμές:

- η πιθανότητα γνησιότητας που παράγει το μοντέλο,
- ο μέσος βαθμός ομοιότητας της εξεταζόμενης υπογραφής με γνήσιες υπογραφές του ίδιου ατόμου.

Η ομοιότητα υπολογίζεται μέσω της **συνημιτονικής ομοιότητας** μεταξύ των διανυσμάτων χαρακτηριστικών των υπογραφών. Για μία πλαστή υπογραφή, η σύγκριση πραγματοποιείται με τα πέντε διαθέσιμα γνήσια δείγματα του συγκεκριμένου ατόμου. Στην περίπτωση μιας γνήσιας υπογραφής εφαρμόζεται λογική **leave-one-out**, δηλαδή η εξεταζόμενη γνήσια υπογραφή αφαιρείται από το σύνολο αναφοράς και συγκρίνεται με τις υπόλοιπες τέσσερις, ώστε να μην πραγματοποιείται σύγκριση της εικόνας με τον εαυτό της.

Ο τελικός συνδυαστικός βαθμός υπολογίζεται σύμφωνα με τη σχέση:

$$CombinedScore = w_p \cdot Probability + w_s \cdot Similarity$$

όπου w_p είναι η βαρύτητα της πιθανότητας που παράγει το μοντέλο και w_s η βαρύτητα του βαθμού ομοιότητας.

Εξετάστηκαν οι ακόλουθοι τρεις συνδυασμοί:

Πίνακας 3.1: Βαρύτητες

Βαρύτητα μοντέλου	Βαρύτητα ομοιότητας
0,65	0,35
0,50	0,50
0,35	0,65

Οι τρεις αυτοί συνδυασμοί είναι ρητά καθορισμένοι στο πειραματικό script.

Και σε αυτή τη φάση χρησιμοποιήθηκε όριο απόφασης **0,50**. Επιπλέον των μετρικών ακρίβεια ταξινόμησης, θετική προγνωστική αξία, ανάκληση και F1-score, υπολογίστηκαν οι μετρικές **ROC-**

AUC και **μέση ακρίβεια**, καθώς και ο μέσος συνδυαστικός βαθμός για τις γνήσιες και τις πλαστές υπογραφές.

3.4 Πειραματικά αποτελέσματα

Στην παρούσα ενότητα παρουσιάζονται τα αποτελέσματα των δύο φάσεων της πειραματικής μελέτης. Αρχικά εξετάζεται η απόδοση των διαφορετικών συνδυασμών μοντέλων βαθιάς και μηχανικής μάθησης, με στόχο την επιλογή της καταλληλότερης προσέγγισης για την ταξινόμηση των υπογραφών. Στη συνέχεια παρουσιάζονται τα αποτελέσματα της συνδυαστικής αξιολόγησης, όπου η πιθανότητα γνησιότητας του επιλεγμένου μοντέλου συνδυάζεται με τον βαθμό ομοιότητας προς τα γνήσια δείγματα, ώστε να προσδιοριστεί η καταλληλότερη προεπιλεγμένη βαρύτητα για τη λειτουργία της εφαρμογής Signalyze.

3.4.1 Αποτελέσματα αξιολόγησης μοντέλων

Η πρώτη φάση της πειραματικής μελέτης παρουσίασε σημαντικές διαφοροποιήσεις μεταξύ των 32 συνδυασμών που εξετάστηκαν.

Ο συνδυασμός **DenseNet169 – Naive Bayes** παρουσίασε την υψηλότερη τιμή ακρίβειας ταξινόμησης, ίση με **58,57%**, με θετική προγνωστική αξία **58,18%**, ανάκληση **60,95%** και F1-βαθμός **59,53%**. Ο πίνακας σύγκρισής του περιλάμβανε 64 True Positives, 59 True Negatives, 46 False Positives και 41 False Negatives.

Ο δεύτερος καλύτερος συνδυασμός ως προς ακρίβεια ταξινόμησης ήταν ο **DenseNet121 – Decision Tree**, με **55,24%**, ενώ οι συνδυασμοί **DenseNet169 – Random Forest** και **DenseNet201 – Naive Bayes** παρουσίασαν ακρίβεια ταξινόμησης **53,33%**.

Παρόλα αυτά, η εξέταση μόνο της ακρίβειας δεν αποδίδει πλήρως τη συμπεριφορά των μοντέλων. Χαρακτηριστικό παράδειγμα αποτελεί ο συνδυασμός **DenseNet169 – Logistic Regression**, ο οποίος πέτυχε ανάκληση **96,19%**, αλλά θετική προγνωστική αξία μόλις **51,27%**. Από τον πίνακα σύγκρισής του προκύπτει ότι αναγνώρισε σωστά 101 από τις 105 γνήσιες υπογραφές, αλλά παράλληλα χαρακτήρισε λανθασμένα 96 από τις 105 πλαστές ως γνήσιες.

Ακόμη πιο χαρακτηριστική είναι η περίπτωση του **MobileNetV2 – Random Forest**, το οποίο πέτυχε ανάκληση **100%**, καθώς αναγνώρισε όλες τις γνήσιες υπογραφές. Ωστόσο, χαρακτήρισε και τις 105 πλαστές υπογραφές ως γνήσιες, με αποτέλεσμα να παρουσιάσει ακρίβεια ταξινόμησης μόλις **50%**.

Τα αποτελέσματα αυτά αναδεικνύουν τη σημασία της ταυτόχρονης εξέτασης πολλαπλών μετρικών και του πίνακα σύγκρισης. Με βάση τη συνολική εικόνα των αποτελεσμάτων και κυρίως την υψηλότερη ακρίβεια ταξινόμησης, ο συνδυασμός **DenseNet169 – Naive Bayes** επιλέχθηκε για την επόμενη φάση της πειραματικής διαδικασίας.

Πίνακας 3.2: Αξιολογήσεις μοντέλων

Μοντέλο	Ακρίβεια ταξινόμησης	Θετική προγνωστική αξία	Ανάκληση	F1-score	TP/TN/FP/FN
DenseNet169_Naive Bayes_model.pkl	58.57%	58.18%	60.95%	59.53%	64/59/46/41
DenseNet121_Decision Tree_model.pkl	55.24%	58.46%	36.19%	44.71%	38/78/27/67

DenseNet169_Random Forest_model.pkl	53.33%	53.85%	46.67%	50.00%	49/63/42/56
DenseNet201_Naive Bayes_model.pkl	53.33%	52.12%	81.90%	63.70%	86/26/79/19
DenseNet169_Logistic Regression_model.pkl	52.38%	51.27%	96.19%	66.89%	101/9/96/4
DenseNet201_Logistic Regression_model.pkl	51.90%	51.37%	71.43%	59.76%	75/34/71/30
VGG16_Naive Bayes_model.pkl	51.90%	60.00%	11.43%	19.20%	12/97/8/93
VGG19_Decision Tree_model.pkl	51.90%	52.22%	44.76%	48.21%	47/62/43/58
DenseNet121_Random Forest_model.pkl	51.43%	54.84%	16.19%	25.00%	17/91/14/88
DenseNet201_Decision Tree_model.pkl	51.43%	51.49%	49.52%	50.49%	52/56/49/53
VGG19_Random Forest_model.pkl	51.43%	50.76%	95.24%	66.23%	100/8/97/5
DenseNet169_Decision Tree_model.pkl	50.95%	51.28%	38.10%	43.72%	40/67/38/65
MobileNet_Naive Bayes_model.pkl	50.95%	53.12%	16.19%	24.82%	17/90/15/88
VGG19_Naive Bayes_model.pkl	50.48%	100.00%	0.95%	1.89%	1/105/0/104
DenseNet121_Naive Bayes_model.pkl	50.00%	50.00%	99.05%	66.45%	104/1/104/1
MobileNet_Decision Tree_model.pkl	50.00%	50.00%	65.71%	56.79%	69/36/69/36
MobileNetV2_Random Forest_model.pkl	50.00%	50.00%	100.00%	66.67%	105/0/105/0
VGG16_Logistic Regression_model.pkl	50.00%	50.00%	92.38%	64.88%	97/8/97/8
VGG19_Logistic Regression_model.pkl	50.00%	50.00%	87.62%	63.67%	92/13/92/13
DenseNet201_Random Forest_model.pkl	49.52%	49.74%	89.52%	63.95%	94/10/95/11
MobileNetV2_Naive Bayes_model.pkl	49.52%	49.72%	85.71%	62.94%	90/14/91/15
DenseNet121_Logistic Regression_model.pkl	48.57%	47.62%	28.57%	35.71%	30/72/33/75

MobileNetV2_Decision Tree_model.pkl	48.57%	48.92%	64.76%	55.74%	68/34/71/37
VGG16_Random Forest_model.pkl	48.57%	41.18%	6.67%	11.48%	7/95/10/98
MobileNetV2_Logistic Regression_model.pkl	48.10%	48.95%	88.57%	63.05%	93/8/97/12
EfficientNetV2B0_Random Forest_model.pkl	47.14%	48.17%	75.24%	58.74%	79/20/85/26
MobileNet_Logistic Regression_model.pkl	47.14%	48.45%	89.52%	62.88%	94/5/100/11
MobileNet_Random Forest_model.pkl	47.14%	48.00%	68.57%	56.47%	72/27/78/33
EfficientNetV2B0_Decision Tree_model.pkl	46.19%	47.65%	77.14%	58.91%	81/16/89/24
VGG16_Decision Tree_model.pkl	43.81%	35.56%	15.24%	21.33%	16/76/29/89
EfficientNetV2B0_Logistic Regression_model.pkl	42.86%	31.71%	12.38%	17.81%	13/77/28/92
EfficientNetV2B0_Naive Bayes_model.pkl	41.90%	37.31%	23.81%	29.07%	25/63/42/80

3.4.2 Αποτελέσματα αξιολόγησης βαρών

Στη δεύτερη φάση εξετάστηκε η επίδραση διαφορετικών βαρών στον συνδυασμό της πιθανότητας που παράγει το μοντέλο DenseNet169 – Naive Bayes και της μέσης ομοιότητας με τα γνήσια δείγματα.

Τα αποτελέσματα παρουσιάζονται στον ακόλουθο πίνακα:

Πίνακας 3.3: Αποτελέσματα αξιολόγησης βαρών

Βαρύτητα μοντέλου	Βαρύτητα ομοιότητας	Ακρίβεια ταξινόμησης	Θετική προγνωστική αξία	Ανάκληση	F1-score	ROC-AUC	Μέση ακρίβεια
0,65	0,35	58,57%	58,18%	60,95%	59,53%	78,10%	81,32%
0,50	0,50	58,57%	58,18%	60,95%	59,53%	78,10%	81,32%
0,35	0,65	62,38%	60,66%	70,48%	65,20%	78,10%	81,32%

Οι δύο πρώτοι συνδυασμοί, δηλαδή βάρος μοντέλου 65% και ομοιότητας 35%, καθώς και βάρος μοντέλου 50% και ομοιότητας 50%, οδήγησαν στην ίδια τελική ταξινόμηση των 210 δειγμάτων, με ακρίβεια 58,57%. Παρότι οι αριθμητικές τιμές των συνδυαστικών βαθμολογιών διέφεραν, καμία

μεταβολή δεν ήταν αρκετή ώστε να μετακινήσει κάποιο δείγμα από τη μία πλευρά του ορίου 0,50 στην άλλη.

Ωστόσο, ο συνδυασμός με βάρος μοντέλου 35% και βάρος ομοιότητας 65% αποδείχθηκε αποτελεσματικότερος. Η ακρίβεια ταξινόμησης αυξήθηκε από 58,57% σε 62,38%, ενώ ο F1-βαθμός αυξήθηκε από 59,53% σε 65,20%. Επιπλέον, η ανάκληση αυξήθηκε σημαντικά, από 60,95% σε 70,48%. Παράλληλα, η θετική προγνωστική αξία παρουσίασε μικρή αύξηση, από 58,18% σε 60,66%.

Με τον συνδυασμό 35/65 αναγνωρίστηκαν σωστά 74 γνήσιες και 57 πλαστές υπογραφές, ενώ καταγράφηκαν 48 ψευδώς θετικά και 31 ψευδώς αρνητικά αποτελέσματα. Σε σύγκριση με τον συνδυασμό ίσων βαρών 50/50, η αλλαγή σε βάρη 35/65 μετέβαλε την τελική απόφαση για 12 από τα 210 δείγματα. Το script υπολόγισε μέση απόλυτη μεταβολή 0,071796 και μέγιστη μεταβολή 0,124193 στον συνδυαστικό βαθμό.

Οι τιμές ROC-AUC (78,10%) και μέση ακρίβεια (81,32%) παρέμειναν ίδιες και στους τρεις συνδυασμούς βαρών, ενώ μεταβλήθηκαν οι μετρικές που εξαρτώνται από το όριο απόφασης 0,50. Αυτό δείχνει ότι η αλλαγή των βαρών δεν επηρέασε τη συνολική ικανότητα κατάταξης των δειγμάτων, αλλά μετακίνησε τις τελικές βαθμολογίες σε σχέση με το συγκεκριμένο όριο και, συνεπώς, άλλαξε την τελική ταξινόμηση ορισμένων υπογραφών.

Επιπλέον, παρατηρείται διαφορά στους μέσους όρους των δύο κατηγοριών. Για βάρη 35/65, ο μέσος συνδυαστικός βαθμός των γνήσιων υπογραφών ήταν 0,6926, ενώ των πλαστών υπογραφών ήταν 0,5579. Παρότι οι μέσες τιμές των δύο κατηγοριών παραμένουν σχετικά κοντά, η αυξημένη βαρύτητα της ομοιότητας οδήγησε στην καλύτερη ταξινόμηση μεταξύ των τριών συνδυασμών που εξετάστηκαν.

Με βάση τα αποτελέσματα, επιλέχθηκαν ως προεπιλεγμένες τιμές βάρος 35% για την πιθανότητα του μοντέλου και 65% για τον βαθμό ομοιότητας. Η επιλογή αυτή βασίστηκε στην υψηλότερη ακρίβεια ταξινόμησης, στη μεγαλύτερη βαθμολογία F1 και στη βελτιωμένη ισορροπία μεταξύ θετικής προγνωστικής αξίας και ανάκλησης στο συγκεκριμένο πειραματικό σύνολο. Ωστόσο, οι τιμές αυτές χρησιμοποιούνται μόνο ως προεπιλεγμένη ρύθμιση, καθώς ο χρήστης μπορεί να μεταβάλλει τη σχετική βαρύτητα των δύο συνιστωσών.

Κεφάλαιο 4ο: Γλώσσες και τεχνολογίες

Για τη δημιουργία της Signalyze χρησιμοποιήθηκαν τεχνολογίες τόσο για την ανάπτυξη της διαδικτυακής εφαρμογής όσο και για το υποσύστημα ανάλυσης υπογραφών. Το back-end υλοποιήθηκε με PHP και CodeIgniter 4, ενώ για την αποθήκευση των δεδομένων χρησιμοποιήθηκε MySQL. Η διεπαφή χρήστη αναπτύχθηκε με HTML, CSS, Bootstrap και JavaScript. Παράλληλα, το υποσύστημα μηχανικής μάθησης υλοποιήθηκε σε Python με βιβλιοθήκες όπως TensorFlow, Keras, scikit-learn και OpenCV. Στις επόμενες ενότητες παρουσιάζονται συνοπτικά οι βασικές τεχνολογίες του έργου.

4.1 Backend

Η κύρια γλώσσα προγραμματισμού του backend είναι η PHP, μέσω της οποίας υλοποιείται η βασική λογική της εφαρμογής, η διαχείριση των αιτημάτων του χρήστη, η επικοινωνία με τη βάση δεδομένων και η διαχείριση των δεδομένων που αφορούν τις υπογραφές.

Το βασικό framework που χρησιμοποιήθηκε είναι το CodeIgniter 4. Πρόκειται για PHP framework που ακολουθεί το αρχιτεκτονικό πρότυπο Model–View–Controller (MVC), διαχωρίζοντας τη λογική της εφαρμογής, τη διαχείριση των δεδομένων και την παρουσίαση της διεπαφής [19]. Ο συγκεκριμένος διαχωρισμός βοηθά στη διατήρηση οργανωμένου κώδικα και διευκολύνει τόσο την ανάπτυξη όσο και τη μελλοντική συντήρηση του συστήματος.

Στη Signalyze, οι controllers αναλαμβάνουν τη διαχείριση των ενεργειών του χρήστη, τα models χρησιμοποιούνται για την επικοινωνία με τη βάση δεδομένων και τα views για τη δημιουργία των σελίδων που εμφανίζονται στον browser. Παράλληλα, το CodeIgniter παρέχει έτοιμες λειτουργίες για routing, validation, sessions, uploads, database access και migrations, περιορίζοντας την ανάγκη για επαναλαμβανόμενο κώδικα και προσφέροντας μια πιο σαφή δομή στην εφαρμογή [19].

Για την αυθεντικοποίηση και την εξουσιοδότηση χρησιμοποιήθηκε το CodeIgniter Shield, το επίσημο authentication framework του CodeIgniter 4 [20]. Μέσω αυτού υποστηρίζεται η δημιουργία λογαριασμών και η σύνδεση χρηστών με email και κωδικό πρόσβασης. Επιπλέον, η εφαρμογή υποστηρίζει σύνδεση μέσω Google OAuth 2.0, ώστε ο χρήστης να μπορεί να χρησιμοποιήσει τον λογαριασμό Google του για την είσοδό του [22].

Για τη διαχείριση των PHP πακέτων χρησιμοποιείται το Composer, ενώ τα PHPUnit και Faker υποστηρίζουν τη δημιουργία αυτοματοποιημένων tests και δοκιμαστικών δεδομένων αντίστοιχα.

Τέλος, η εφαρμογή μπορεί να αξιοποιήσει το OpenAI API ως εναλλακτικό μηχανισμό ανάλυσης μιας υπογραφής, μέσω του API key που ορίζει ο ίδιος ο χρήστης.

4.2 Βάση δεδομένων

Για τη μόνιμη αποθήκευση των δεδομένων χρησιμοποιείται MySQL/MariaDB. Η εφαρμογή επικοινωνεί με τη βάση δεδομένων μέσω του MySQLi driver που υποστηρίζεται από το CodeIgniter.

Στη βάση αποθηκεύονται οι πληροφορίες που αφορούν τους χρήστες, τα πρόσθετα μέλη, τις γνήσιες υπογραφές αναφοράς και τα υπόλοιπα δεδομένα που απαιτούνται για τη λειτουργία της εφαρμογής.

Η δημιουργία και η εξέλιξη της δομής της βάσης πραγματοποιείται με τα CodeIgniter Migrations, τα οποία επιτρέπουν την οργανωμένη διαχείριση των αλλαγών στους πίνακες και στα πεδία. Αντίστοιχα, τα Seeds μπορούν να χρησιμοποιηθούν για την εισαγωγή αρχικών ή δοκιμαστικών δεδομένων.

Για την εκτέλεση αυτοματοποιημένων tests έχει επίσης διαμορφωθεί SQLite in-memory βάση δεδομένων, ώστε οι δοκιμές να μην επηρεάζουν τα πραγματικά δεδομένα της εφαρμογής.

4.3 Frontend

Η διεπαφή χρήστη της Signalyze αναπτύχθηκε με HTML5, CSS3 και JavaScript. Η HTML χρησιμοποιείται για τη δομή των σελίδων, η CSS για τη μορφοποίηση και η JavaScript για τις λειτουργίες που απαιτούν αλληλεπίδραση στην πλευρά του browser.

Για τη σχεδίαση της διεπαφής χρησιμοποιήθηκε το Bootstrap 5.3.3, το οποίο παρέχει έτοιμα responsive στοιχεία, όπως φόρμες, πίνακες, navigation menus, buttons, alerts και modal παράθυρα [32]. Με τον τρόπο αυτό η εφαρμογή μπορεί να προσαρμόζεται ευκολότερα σε διαφορετικές αναλύσεις οθόνης και σε φορητές συσκευές.

Οι τελικές σελίδες παράγονται κυρίως μέσω των PHP Views του CodeIgniter, επομένως η εφαρμογή βασίζεται κατά κύριο λόγο σε server-side rendering.

4.4 Μηχανική Μάθηση και επεξεργασία υπογραφών

Το υποσύστημα ανάλυσης των υπογραφών αναπτύχθηκε σε Python 3. Η Python χρησιμοποιήθηκε για την εκτέλεση των scripts που αφορούν την εξαγωγή χαρακτηριστικών, την ταξινόμηση των υπογραφών και την πειραματική αξιολόγηση των διαφορετικών μοντέλων.

Για τη βαθιά μάθηση χρησιμοποιήθηκαν τα TensorFlow και Keras [24]. Μέσω αυτών φορτώθηκαν προεκπαιδευμένα συνελκτικά νευρωνικά δίκτυα, τα οποία χρησιμοποιήθηκαν ως εξαγωγείς χαρακτηριστικών από τις εικόνες των υπογραφών.

Στην πειραματική μελέτη εξετάστηκαν οι αρχιτεκτονικές DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, MobileNet, MobileNetV2, VGG16 και VGG19. Ιδιαίτερη σημασία έχει η οικογένεια DenseNet, η οποία χαρακτηρίζεται από πυκνές συνδέσεις μεταξύ των επιπέδων και έχει σχεδιαστεί ώστε να ενισχύει την επαναχρησιμοποίηση των χαρακτηριστικών κατά τη διάδοση της πληροφορίας στο δίκτυο [26].

Για την τελική ταξινόμηση των χαρακτηριστικών χρησιμοποιήθηκε η βιβλιοθήκη scikit-learn [25]. Στο πλαίσιο της εργασίας εξετάστηκαν οι αλγόριθμοι Naive Bayes, Random Forest, Logistic Regression και Decision Tree, όπως παρουσιάστηκε αναλυτικότερα στην πειραματική μελέτη του Κεφαλαίου 3.

Η OpenCV χρησιμοποιήθηκε για τη φόρτωση και την προεπεξεργασία των εικόνων των υπογραφών, όπως η αλλαγή διαστάσεων και η μετατροπή χρωματικού χώρου, ενώ η NumPy χρησιμοποιήθηκε για τη διαχείριση των αριθμητικών πινάκων και των διανυσμάτων χαρακτηριστικών.

Επιπλέον, για τη σύγκριση μιας εξεταζόμενης υπογραφής με τις αποθηκευμένες γνήσιες υπογραφές χρησιμοποιείται η **συνημιτονική ομοιότητα**. Η τιμή ομοιότητας συνδυάζεται με την πιθανότητα που παράγει το μοντέλο ώστε να προκύψει ο τελικός βαθμός γνησιότητας.

Τα εκπαιδευμένα μοντέλα αποθηκεύονται σε αρχεία .pkl όταν πρόκειται για κλασικούς ταξινομητές μηχανικής μάθησης, όπως το Random Forest. Για μοντέλα βαθιάς μάθησης που υλοποιούνται με Keras/TensorFlow, χρησιμοποιούνται οι μορφές .keras ή .h5, ώστε να μπορούν να φορτωθούν και να επαναχρησιμοποιηθούν χωρίς να απαιτείται νέα εκπαίδευση.

Η εφαρμογή έχει σχεδιαστεί ώστε να εκτελείται σε PHP-compatible web server, με το public directory του CodeIgniter να αποτελεί το δημόσια προσβάσιμο σημείο της εφαρμογής.

Για την ανάπτυξη του έργου χρησιμοποιήθηκε το **Visual Studio Code (VS Code)** ως κύριο περιβάλλον συγγραφής και επεξεργασίας του πηγαίου κώδικα. Το VS Code παρέχει υποστήριξη για διαφορετικές γλώσσες προγραμματισμού και επεκτάσεις, γεγονός που το καθιστά κατάλληλο για ένα έργο όπως η Signalyze, στο οποίο συνυπάρχουν PHP, Python, HTML, CSS και JavaScript.

Για τη διαχείριση λειτουργιών του CodeIgniter χρησιμοποιείται το **Spark CLI**, μέσω του οποίου μπορούν να εκτελούνται εντολές σχετικές με migrations, database seeds, cache και άλλες εργασίες ανάπτυξης. Οι ρυθμίσεις του περιβάλλοντος, όπως τα στοιχεία σύνδεσης στη βάση δεδομένων και τα OAuth credentials, αποθηκεύονται στο αρχείο .env, ώστε να διαχωρίζονται από τον βασικό πηγαίο κώδικα.

Για τη δημιουργία και διαχείριση του τοπικού περιβάλλοντος ανάπτυξης χρησιμοποιήθηκε το **MAMP PRO**. Μέσω αυτού ήταν δυνατή η τοπική εκτέλεση της εφαρμογής με web server, PHP και MySQL, καθώς και η διαχείριση των επιμέρους ρυθμίσεων του project. Η χρήση του MAMP PRO διευκόλυνε την ανάπτυξη, τη δοκιμή και τον έλεγχο της λειτουργίας της Signalyze σε τοπικό περιβάλλον [33].

Κεφάλαιο 5ο: Σχεδίαση και Υλοποίηση της Signalyze

Στο παρόν κεφάλαιο παρουσιάζεται η διαδικασία σχεδίασης και υλοποίησης της εφαρμογής Signalyze. Αρχικά καταγράφονται οι βασικές λειτουργικές απαιτήσεις του συστήματος με τη μορφή user stories. Στη συνέχεια παρουσιάζεται η αρχιτεκτονική της εφαρμογής και οι κύριες περιπτώσεις χρήσης. Ακολουθεί η σχεδίαση της βάσης δεδομένων, μέσω διαγράμματος οντοτήτων–συσχετίσεων και σχεσιακού σχήματος. Τέλος, περιγράφονται η υλοποίηση του backend και του frontend, καθώς και η οργάνωση του πηγαίου κώδικα στο αποθετήριο GitHub.

5.1 Λειτουργικές απαιτήσεις

Οι λειτουργικές απαιτήσεις περιγράφουν τις βασικές δυνατότητες που πρέπει να παρέχει το σύστημα στους χρήστες του. Στην περίπτωση της Signalyze, οι απαιτήσεις καταγράφηκαν με τη μορφή user stories, ώστε κάθε λειτουργία να αποτυπώνεται από την οπτική του τελικού χρήστη και να συνδέεται με έναν συγκεκριμένο στόχο. Με αυτόν τον τρόπο γίνεται σαφέστερο ποιες ανάγκες καλύπτει η εφαρμογή και πώς αυτές μεταφράζονται σε επιμέρους λειτουργίες του συστήματος.

Πίνακας 5.1: Λειτουργικές απαιτήσεις

ID	User Story
US01	Ως επισκέπτης, θέλω να μπορώ να δημιουργήσω λογαριασμό ώστε να αποκτήσω πρόσβαση στις λειτουργίες της εφαρμογής.
US02	Ως χρήστης, θέλω να μπορώ να συνδεθώ στην εφαρμογή με email και κωδικό πρόσβασης.
US03	Ως χρήστης, θέλω να μπορώ να συνδεθώ μέσω του λογαριασμού μου Google, ώστε η διαδικασία εισόδου να είναι πιο άμεση.
US04	Ως χρήστης, θέλω να μπορώ να καταχωρίζω γνήσιες υπογραφές αναφοράς, ώστε να δημιουργείται το προσωπικό μου σύνολο δειγμάτων.
US05	Ως χρήστης, θέλω να μπορώ να υποβάλλω μία νέα υπογραφή προς έλεγχο μέσω αρχείου εικόνας.
US06	Ως χρήστης, θέλω να μπορώ να φωτογραφίζω μία υπογραφή μέσω κινητής συσκευής και να τη χρησιμοποιώ άμεσα για έλεγχο.
US07	Ως χρήστης, θέλω να λαμβάνω αποτέλεσμα σχετικά με το αν μία υπογραφή εκτιμάται ως γνήσια ή πλαστή.
US08	Ως χρήστης, θέλω η αξιολόγηση να λαμβάνει υπόψη τόσο το αποτέλεσμα του μοντέλου όσο και την ομοιότητα με τις γνήσιες υπογραφές αναφοράς.
US09	Ως χρήστης, θέλω να μπορώ να προσαρμόζω τη βαρύτητα μεταξύ του αποτελέσματος του μοντέλου και του βαθμού ομοιότητας, ώστε να επηρεάζω τον τρόπο διαμόρφωσης του τελικού αποτελέσματος.
US10	Ως χρήστης, θέλω να μπορώ να χρησιμοποιώ εναλλακτική αξιολόγηση μέσω του OpenAI API.

US11	Ως χρήστης, θέλω να μπορώ να δημιουργώ έως και πέντε πρόσθετα μέλη στον λογαριασμό μου.
US12	Ως χρήστης, θέλω κάθε πρόσθετο μέλος να διαθέτει το δικό του ανεξάρτητο σύνολο γνήσιων υπογραφών αναφοράς.
US13	Ως χρήστης, θέλω να μπορώ να πραγματοποιώ ανεξάρτητους ελέγχους υπογραφών για κάθε πρόσθετο μέλος.
US14	Ως χρήστης, θέλω να μπορώ να διαχειρίζομαι τις αποθηκευμένες γνήσιες υπογραφές μου, όπως να προσθέτω ή να διαγράφω δείγματα.
US15	Ως χρήστης, θέλω να μπορώ να αποσυνδέομαι με ασφάλεια από την εφαρμογή.

Συνολικά, οι παραπάνω λειτουργικές απαιτήσεις καλύπτουν τις βασικές λειτουργίες της Signalyze, από τη διαχείριση του λογαριασμού και των γνήσιων δειγμάτων έως την εκτέλεση ελέγχων γνησιότητας και τη διαχείριση πρόσθετων μελών. Οι απαιτήσεις αυτές αποτέλεσαν τη βάση για τον σχεδιασμό της αρχιτεκτονικής και των περιπτώσεων χρήσης της εφαρμογής, οι οποίες παρουσιάζονται στην επόμενη ενότητα.

5.2 Αρχιτεκτονική της Εφαρμογής (use cases)

Στην παρούσα ενότητα παρουσιάζεται η λειτουργική αρχιτεκτονική της Signalyze μέσα από τις βασικές περιπτώσεις χρήσης (use cases). Σκοπός είναι να αποτυπωθούν οι κύριες δυνατότητες που παρέχει η εφαρμογή και ο τρόπος με τον οποίο ο χρήστης αλληλεπιδρά με αυτές. Οι περιπτώσεις χρήσης προκύπτουν από τις λειτουργικές απαιτήσεις που παρουσιάστηκαν στην προηγούμενη ενότητα και περιγράφουν τις βασικές ενέργειες που μπορεί να πραγματοποιήσει ένας επισκέπτης ή ένας εγγεγραμμένος χρήστης.

5.2.1 Διάγραμμα περιπτώσεων χρήσης

Για τη συνοπτική απεικόνιση των λειτουργιών της εφαρμογής χρησιμοποιείται διάγραμμα περιπτώσεων χρήσης (Use Case Diagram). Στο διάγραμμα παρουσιάζονται οι βασικοί actors του συστήματος και οι λειτουργίες με τις οποίες αυτοί μπορούν να αλληλεπιδράσουν.

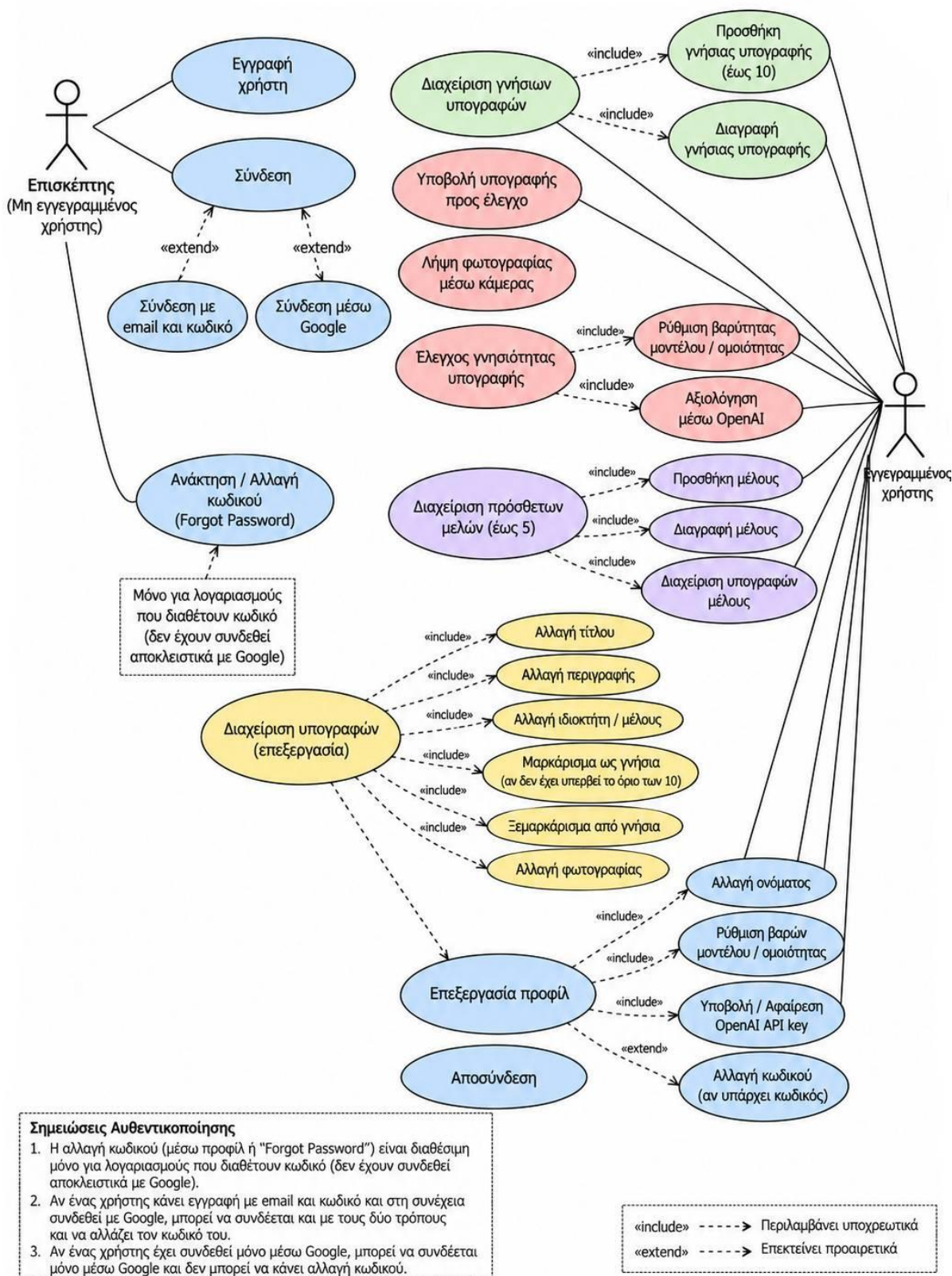
Οι δύο βασικές κατηγορίες χρηστών είναι ο επισκέπτης και ο εγγεγραμμένος χρήστης. Ο επισκέπτης έχει τη δυνατότητα να δημιουργήσει νέο λογαριασμό ή να συνδεθεί στην εφαρμογή είτε μέσω email και κωδικού πρόσβασης είτε μέσω λογαριασμού Google. Επιπλέον, μπορεί να χρησιμοποιήσει τη λειτουργία ανάκτησης κωδικού, εφόσον ο λογαριασμός του διαθέτει κωδικό πρόσβασης. Μετά την επιτυχή αυθεντικοποίηση, ο εγγεγραμμένος χρήστης αποκτά πρόσβαση στις κύριες λειτουργίες της Signalyze.

Οι βασικές περιπτώσεις χρήσης που αποτυπώνονται στο διάγραμμα είναι:

- Εγγραφή χρήστη
- Σύνδεση με email και κωδικό πρόσβασης
- Σύνδεση μέσω Google
- Ανάκτηση ή αλλαγή κωδικού πρόσβασης
- Διαχείριση γνήσιων υπογραφών
- Επεξεργασία υπογραφής
- Αλλαγή τίτλου και περιγραφής υπογραφής

- Αλλαγή ιδιοκτήτη ή μέλους στο οποίο ανήκει η υπογραφή
- Χαρακτηρισμός ή αποχαρακτηρισμός μιας υπογραφής ως γνήσιας
- Αλλαγή της εικόνας μιας υπογραφής
- Υποβολή υπογραφής προς έλεγχο
- Λήψη φωτογραφίας υπογραφής μέσω κάμερας
- Έλεγχος γνησιότητας υπογραφής
- Ρύθμιση της βαρύτητας μεταξύ μοντέλου και ομοιότητας
- Εναλλακτική αξιολόγηση μέσω OpenAI
- Διαχείριση πρόσθετων μελών
- Διαχείριση γνήσιων υπογραφών ενός μέλους
- Έλεγχος υπογραφής ενός μέλους
- Επεξεργασία προφίλ
- Αλλαγή ονόματος χρήστη
- Υποβολή ή αφαίρεση OpenAI API key
- Αλλαγή κωδικού πρόσβασης μέσω προφίλ, όπου αυτή υποστηρίζεται
- Αποσύνδεση από την εφαρμογή

Οι δυνατότητες που σχετίζονται με την αλλαγή ή την ανάκτηση κωδικού διαφοροποιούνται ανάλογα με τον τρόπο δημιουργίας και αυθεντικοποίησης του λογαριασμού, όπως αποτυπώνεται και στο διάγραμμα.



Σχήμα 5.2.1: Διάγραμμα περιπτώσεων χρήσης

5.2.2 Βασικές περιπτώσεις χρήσης

Οι βασικές περιπτώσεις χρήσης της εφαρμογής Signalyze αφορούν τις λειτουργίες που εκτελεί ο χρήστης για τη διαχείριση και την ανάλυση χειρόγραφων υπογραφών. Η κύρια λειτουργία της εφαρμογής είναι ο έλεγχος μιας νέας υπογραφής, με σκοπό να προσδιοριστεί εάν αυτή είναι γνήσια ή πλαστή. Για την πραγματοποίηση του ελέγχου, η υπογραφή που εισάγει ο χρήστης συγκρίνεται με το σύνολο των γνήσιων υπογραφών αναφοράς που έχουν αποθηκευτεί προηγουμένως στο σύστημα.

Μία ακόμη βασική περίπτωση χρήσης είναι η διαχείριση των υπογραφών αναφοράς. Ο χρήστης έχει τη δυνατότητα να προσθέτει και να διαχειρίζεται τις γνήσιες υπογραφές που χρησιμοποιούνται ως βάση σύγκρισης κατά τη διαδικασία ανάλυσης. Οι υπογραφές αυτές αποτελούν το προσωπικό σύνολο αναφοράς του χρήστη και χρησιμοποιούνται από το σύστημα για τον υπολογισμό της ομοιότητας με κάθε νέα υπογραφή που υποβάλλεται προς έλεγχο.

Η εφαρμογή παρέχει επίσης τη δυνατότητα παραμετροποίησης της διαδικασίας ανάλυσης. Ο χρήστης μπορεί να ρυθμίσει τους συντελεστές βαρύτητας που εφαρμόζονται στην πιθανότητα ταξινόμησης του μοντέλου μηχανικής μάθησης και στην συνημιτονική ομοιότητα. Με αυτόν τον τρόπο μπορεί να επηρεάσει τον βαθμό συμμετοχής κάθε μεθόδου στο τελικό αποτέλεσμα της ανάλυσης.

Παράλληλα, υπάρχει η δυνατότητα προαιρετικής αξιολόγησης της υπογραφής μέσω του OpenAI. Η λειτουργία αυτή μπορεί να ενεργοποιηθεί από τον χρήστη και λειτουργεί συμπληρωματικά προς την κύρια διαδικασία ανάλυσης, παρέχοντας επιπλέον πληροφορίες σχετικά με το αποτέλεσμα.

Τέλος, η Signalyze υποστηρίζει τη διαχείριση επιπλέον μελών από τον ίδιο λογαριασμό χρήστη. Κάθε χρήστης μπορεί να δημιουργήσει έως και πέντε επιπλέον μέλη, καθένα από τα οποία διαθέτει ανεξάρτητο σύνολο γνήσιων υπογραφών αναφοράς. Με αυτόν τον τρόπο, ένας λογαριασμός μπορεί να χρησιμοποιηθεί για την ανάλυση υπογραφών διαφορετικών ατόμων, διατηρώντας ξεχωριστά τα δεδομένα και τις υπογραφές αναφοράς κάθε μέλους.

5.2.3 Περιγραφή κύριας περίπτωσης χρήσης

Η κύρια περίπτωση χρήσης της Signalyze αφορά τον έλεγχο γνησιότητας μιας υπογραφής. Η συγκεκριμένη διαδικασία συγκεντρώνει τις βασικές λειτουργίες της εφαρμογής, καθώς προϋποθέτει την ύπαρξη γνήσιων υπογραφών αναφοράς, την εισαγωγή μιας νέας υπογραφής προς αξιολόγηση και την εκτέλεση του αντίστοιχου μηχανισμού ελέγχου. Στον ακόλουθο πίνακα παρουσιάζονται συνοπτικά τα βασικά στοιχεία της συγκεκριμένης περίπτωσης χρήσης, όπως ο actor, οι προϋποθέσεις, η είσοδος, η κύρια ροή και το τελικό αποτέλεσμα.

Πίνακας 5.2: Περιγραφή της κύριας περίπτωσης χρήσης

Στοιχείο	Περιγραφή
Περίπτωση χρήσης	Έλεγχος γνησιότητας υπογραφής
Actor	Εγγεγραμμένος χρήστης
Προϋπόθεση	Ο χρήστης έχει συνδεθεί και έχει καταχωρίσει τον απαιτούμενο αριθμό γνήσιων υπογραφών.
Είσοδος	Εικόνα υπογραφής μέσω upload ή κάμερας.

Κύρια ροή	Ο χρήστης εισάγει την υπογραφή, επιλέγει τις επιθυμητές ρυθμίσεις και ξεκινά τον έλεγχο.
Αποτέλεσμα	Η εφαρμογή εμφανίζει την εκτίμηση σχετικά με τη γνησιότητα της υπογραφής.
Εναλλακτική ροή	Ο χρήστης επιλέγει αξιολόγηση μέσω OpenAI.

5.3 Σχεδίαση Βάσης Δεδομένων

Η βάση δεδομένων της Signalyze σχεδιάστηκε ώστε να υποστηρίζει τις βασικές λειτουργίες της εφαρμογής και να διατηρεί οργανωμένα τα δεδομένα που αφορούν τους χρήστες, τα πρόσθετα μέλη και τις υπογραφές. Παράλληλα, στη δομή της ενσωματώνονται οι πίνακες που χρησιμοποιεί το CodeIgniter Shield για την αυθεντικοποίηση και τη διαχείριση των σχετικών στοιχείων ασφαλείας.

Η σχεδίαση της βάσης βασίστηκε στον διαχωρισμό των δεδομένων ανάλογα με τον ρόλο τους στην εφαρμογή. Οι βασικοί πίνακες που αφορούν άμεσα τη λειτουργία της Signalyze είναι οι users, user_members και signature, ενώ οι πίνακες auth_* υποστηρίζουν τις λειτουργίες σύνδεσης, τα identities, τα δικαιώματα και τα tokens των χρηστών. Στις επόμενες υποενότητες παρουσιάζονται το διάγραμμα οντοτήτων-συσχετίσεων και το σχεσιακό σχήμα της βάσης.

5.3.1 Διάγραμμα Οντοτήτων-Συσχετίσεων

Το διάγραμμα οντοτήτων-συσχετίσεων (Entity-Relationship Diagram – ER Diagram) χρησιμοποιείται για τη γραφική απεικόνιση των βασικών οντοτήτων της βάσης δεδομένων και των μεταξύ τους σχέσεων.

Η κεντρική οντότητα της βάσης είναι ο πίνακας users, στον οποίο αποθηκεύονται τα βασικά στοιχεία και οι ρυθμίσεις κάθε χρήστη της εφαρμογής. Μεταξύ αυτών περιλαμβάνονται το όνομα του χρήστη, οι προεπιλεγμένες βαρύτητες που χρησιμοποιούνται κατά την ανάλυση μιας υπογραφής, το OpenAI API key και η πληροφορία σχετικά με το αν ο λογαριασμός υποστηρίζει σύνδεση μέσω κωδικού πρόσβασης.

Ο πίνακας user_members χρησιμοποιείται για την αποθήκευση των πρόσθετων μελών που δημιουργούνται από έναν χρήστη. Η σχέση μεταξύ users και user_members είναι ένα προς πολλά (1:N), καθώς ένας χρήστης μπορεί να διαθέτει περισσότερα μέλη, ενώ κάθε μέλος ανήκει σε έναν συγκεκριμένο χρήστη.

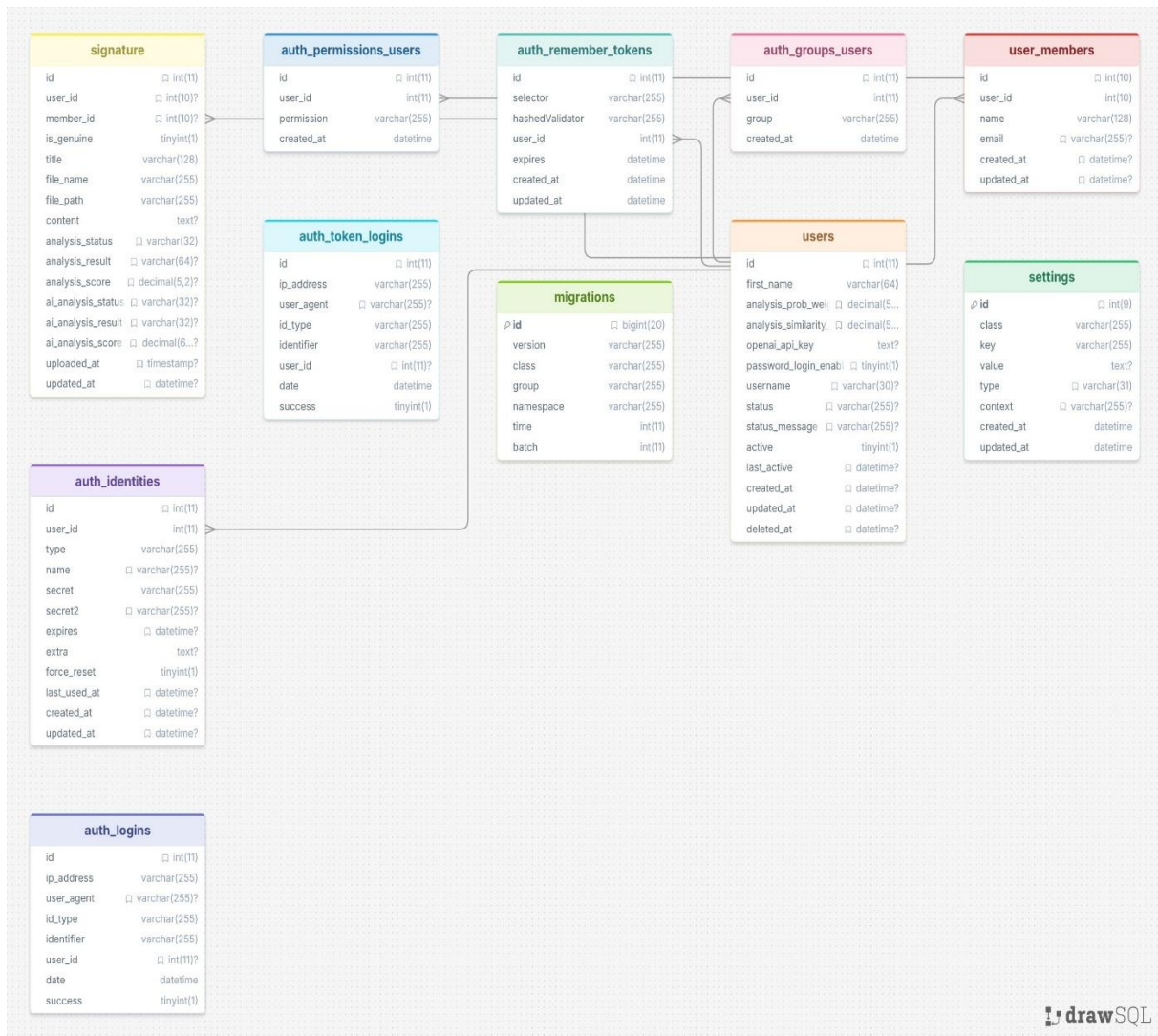
Ο πίνακας signature αποθηκεύει τις υπογραφές που εισάγονται στην εφαρμογή. Για κάθε υπογραφή διατηρούνται στοιχεία όπως ο τίτλος, το όνομα και η διαδρομή του αρχείου, η ένδειξη αν η υπογραφή έχει χαρακτηριστεί ως γνήσια, καθώς και τα αποτελέσματα της βασικής και της εναλλακτικής ανάλυσης. Μέσω του πεδίου member_id μία υπογραφή μπορεί να συσχετιστεί με κάποιο πρόσθετο μέλος. Η σχέση μεταξύ user_members και signature είναι επίσης ένα προς πολλά (1:N).

Σε περίπτωση διαγραφής ενός μέλους, οι υπογραφές που έχουν συνδεθεί με αυτό δεν διαγράφονται. Το member_id των συγκεκριμένων εγγραφών μετατρέπεται σε NULL, σύμφωνα με τον περιορισμό ON DELETE SET NULL που έχει οριστεί στη βάση. Ο πίνακας signature διαθέτει επίσης το πεδίο user_id, μέσω του οποίου συνδέεται η υπογραφή με τον κύριο χρήστη σε επίπεδο εφαρμογής, χωρίς να έχει οριστεί αντίστοιχο foreign key constraint στη βάση.

Παράλληλα, ο πίνακας users συνδέεται με τους βασικούς πίνακες αυθεντικοποίησης του CodeIgniter Shield. Ο πίνακας auth_identities αποθηκεύει τις ταυτότητες αυθεντικοποίησης του χρήστη, ενώ οι

auth_groups_users, auth_permissions_users και auth_remember_tokens χρησιμοποιούνται για ομάδες, δικαιώματα και tokens διατήρησης σύνδεσης αντίστοιχα. Οι συγκεκριμένοι πίνακες συνδέονται με τον users μέσω foreign keys και χρησιμοποιούν ON DELETE CASCADE, ώστε τα σχετικά δεδομένα να διαγράφονται όταν διαγράφεται ο αντίστοιχος χρήστης.

Στο ακόλουθο διάγραμμα παρουσιάζονται οι βασικές οντότητες της βάσης και οι σημαντικότερες σχέσεις μεταξύ τους.



Σχήμα 5.3.1: Διάγραμμα ERD (Entity Relationship Diagram)

5.3.2 Σχεσιακό Σχήμα

Το σχεσιακό σχήμα παρουσιάζει τη δομή των πινάκων της βάσης δεδομένων, τα βασικά πεδία τους και τις μεταξύ τους σχέσεις. Οι κυριότεροι πίνακες της εφαρμογής είναι οι ακόλουθοι:

- **USERS**

id PK, first_name, analysis_prob_weight, analysis_similarity_weight, openai_api_key, password_login_enabled, username, status, status_message, active, last_active, created_at, updated_at, deleted_at

Ο πίνακας users αποθηκεύει τα βασικά στοιχεία του λογαριασμού και τις ρυθμίσεις του χρήστη.

Εκτός από τα γενικά στοιχεία του προφίλ, περιλαμβάνει τις βαρύτητες της πιθανότητας του μοντέλου και της ομοιότητας, το OpenAI API key και την πληροφορία σχετικά με το αν είναι ενεργοποιημένη η σύνδεση μέσω κωδικού.

- **USER_MEMBERS**

id PK, user_id FK → USERS.id, name, email, created_at, updated_at

Ο πίνακας user_members περιλαμβάνει τα πρόσθετα μέλη που δημιουργούνται μέσα από έναν λογαριασμό. Το πεδίο user_id αποτελεί foreign key προς τον πίνακα users, εξασφαλίζοντας ότι κάθε μέλος αντιστοιχεί σε έναν συγκεκριμένο χρήστη.

- **SIGNATURE**

id PK, user_id, member_id FK → USER_MEMBERS.id, is_genuine, title, file_name, file_path, content, analysis_status, analysis_result, analysis_score, ai_analysis_status, ai_analysis_result, ai_analysis_score, uploaded_at, updated_at

Ο πίνακας signature περιλαμβάνει τις υπογραφές που έχουν αποθηκευτεί στην εφαρμογή. Το πεδίο is_genuine δηλώνει αν μία υπογραφή χρησιμοποιείται ως γνήσιο δείγμα αναφοράς, ενώ τα πεδία analysis_status, analysis_result και analysis_score αποθηκεύουν πληροφορίες σχετικά με τη βασική μέθοδο αξιολόγησης. Αντίστοιχα, τα ai_analysis_status, ai_analysis_result και ai_analysis_score αφορούν την εναλλακτική αξιολόγηση μέσω OpenAI.

Για την αυθεντικοποίηση και την εξουσιοδότηση χρησιμοποιούνται επιπλέον οι παρακάτω πίνακες του CodeIgniter Shield:

- **AUTH_IDENTITIES**

id PK, user_id FK → USERS.id, type, name, secret, secret2, expires, extra, force_reset, last_used_at, created_at, updated_at

- **AUTH_GROUPS_USERS**

id PK, user_id FK → USERS.id, group, created_at

- **AUTH_PERMISSIONS_USERS**

id PK, user_id FK → USERS.id, permission, created_at

AUTH_REMEMBER_TOKENS

id PK, user_id FK → USERS.id, selector, hashedValidator, expires, created_at, updated_at

Οι πίνακες auth_logins και auth_token_logins χρησιμοποιούνται για την καταγραφή ενεργειών σύνδεσης. Περιλαμβάνουν στοιχεία όπως τη διεύθυνση IP, το user agent, το αναγνωριστικό που χρησιμοποιήθηκε κατά τη σύνδεση, την ημερομηνία και την ένδειξη επιτυχίας ή αποτυχίας. Παρότι διαθέτουν πεδίο user_id, στο υπάρχον schema δεν έχει οριστεί foreign key constraint προς τον πίνακα users.

Τέλος, οι πίνακες migrations και settings εξυπηρετούν υποστηρικτικές λειτουργίες του framework και της εφαρμογής. Ο πίνακας migrations καταγράφει τις μεταβολές της δομής της βάσης που έχουν εκτελεστεί, ενώ ο settings χρησιμοποιείται για την αποθήκευση γενικότερων ρυθμίσεων.

Συνολικά, η δομή της βάσης δεδομένων διαχωρίζει με σαφή τρόπο τα δεδομένα των χρηστών, των πρόσθετων μελών και των υπογραφών από τα δεδομένα που αφορούν την αυθεντικοποίηση. Παράλληλα, η αξιοποίηση των πινάκων του CodeIgniter Shield επιτρέπει την ενσωμάτωση των λειτουργιών σύνδεσης και διαχείρισης ταυτοτήτων χωρίς την ανάγκη ανάπτυξης ανεξάρτητου μηχανισμού αυθεντικοποίησης.

5.4 Υλοποίηση του Backend

Το backend της Signalyze υλοποιήθηκε με το CodeIgniter 4 και είναι υπεύθυνο για τη διαχείριση των αιτημάτων του χρήστη, την αυθεντικοποίηση, την αποθήκευση και ανάκτηση δεδομένων, καθώς και την επικοινωνία με το υποσύστημα ανάλυσης υπογραφών που έχει αναπτυχθεί σε Python. Η εφαρμογή ακολουθεί την αρχιτεκτονική MVC, με τις routes να αντιστοιχίζουν τα HTTP αιτήματα στις κατάλληλες μεθόδους των controllers και τα models να αναλαμβάνουν την επικοινωνία με τη βάση δεδομένων.

Στην παρούσα ενότητα παρουσιάζονται τα βασικότερα σημεία της υλοποίησης του backend και αντιπροσωπευτικά αποσπάσματα κώδικα. Η παρουσίαση των οθονών και της χρήσης της εφαρμογής πραγματοποιείται ξεχωριστά στο επόμενο κεφάλαιο.

5.4.1 Δρομολόγηση Αιτημάτων

Οι διαδρομές της εφαρμογής ορίζονται στο αρχείο Routes.php και καθορίζουν ποια μέθοδος ενός controller θα εκτελεστεί για κάθε HTTP αίτημα. Για τη διαχείριση των υπογραφών έχουν οριστεί διαφορετικές routes για την προβολή, την εισαγωγή, την ανάλυση και τον χαρακτηρισμό μιας υπογραφής ως γνήσιας. Παράλληλα, υπάρχουν ξεχωριστές διαδρομές για την αυθεντικοποίηση μέσω CodeIgniter Shield και για τη σύνδεση μέσω Google.

Για παράδειγμα, η διαδρομή `signatures/(:num)/analyze` οδηγεί στη μέθοδο `analyze()` του Signatures controller, με το `(:num)` να αντιστοιχεί στο αναγνωριστικό της υπογραφής. Αντίστοιχα, η `analyze-ai` χρησιμοποιείται για την εναλλακτική αξιολόγηση μέσω OpenAI.

```
9 $routes->get('/', 'Home::index');
10 $routes->get('about', 'Home::about');
11 $routes->get('instructions', 'Home::instructions');
12
13 $routes->get('signatures', 'Signatures::index');
14 $routes->get('signatures/training', 'Signatures::trainingUpload');
15 $routes->post('signatures/training/bulk', 'Signatures::bulkCreate');
16 $routes->get('signatures/(:num)', 'Signatures::show/$1');
17 $routes->get('signatures/new', 'Signatures::new', ['as' => 'new_signature']);
18 $routes->post('signatures', 'Signatures::create');
19 $routes->post('signatures/(:num)/analyze', 'Signatures::analyze/$1');
20 $routes->post('signatures/(:num)/analyze-ai', 'Signatures::analyzeAi/$1');
21 $routes->post('signatures/(:num)/toggle-genuine', 'Signatures::toggleGenuine/$1');
22 $routes->get('signatures/(:num)/image', 'Signatures::image/$1');
23 $routes->get("signatures/(:num)/edit/", "Signatures::edit/$1");
24 $routes->patch("signatures/(:num)", "Signatures::update/$1");
25 $routes->delete("signatures/(:num)", "Signatures::delete/$1");
26
27 $routes->get("signatures/(:num)/delete", "Signatures::confirmDelete/$1");
28
29 service('auth')->routes($routes);
30
31 $routes->get('auth/google', 'GoogleAuth::redirect', ['as' => 'google_signin']);
32 $routes->get('auth/google/callback', 'GoogleAuth::callback', ['as' => 'google_callback']);
```

Σχήμα 5.4.1: Βασικές διαδρομές για τη διαχείριση υπογραφών και την αυθεντικοποίηση.

5.4.2 Αυθεντικοποίηση και έλεγχος πρόσβασης

Η αυθεντικοποίηση των χρηστών πραγματοποιείται με το CodeIgniter Shield. Η εφαρμογή χρησιμοποιεί session-based authentication, ώστε μετά την επιτυχημένη σύνδεση να διατηρείται η ταυτότητα του χρήστη στα επόμενα αιτήματα.

Εκτός από τον έλεγχο ύπαρξης ενεργού session, το backend ελέγχει και αν μία συγκεκριμένη υπογραφή ανήκει πράγματι στον συνδεδεμένο χρήστη. Η διαδικασία αυτή είναι σημαντική, καθώς το ID μιας υπογραφής μπορεί να υπάρχει μέσα σε μία URL και δεν πρέπει να είναι δυνατή η πρόσβαση σε υπογραφή διαφορετικού λογαριασμού απλώς με την αλλαγή του ID.

```
16 references
1124 private function currentUserId(): int
1125 {
1126     $user = auth()->user();
1127
1128     return (int) ($user->id ?? 0);
1129 }
1130
9 references
1131 private function assertCanAccessSignature(Signature $signature): void
1132 {
1133     if ((int) ($signature->user_id ?? 0) !== $this->currentUserId()) {
1134         throw new PageNotFoundException('Signature with id ' . $signature->id . ' not found!');
1135     }
1136 }
1137
```

Σχήμα 5.4.2: Έλεγχος σύνδεσης και ιδιοκτησίας υπογραφής

5.4.3 Μεταφόρτωση και αποθήκευση υπογραφών

Η εισαγωγή μιας νέας υπογραφής πραγματοποιείται μέσω της μεθόδου create() του Signatures controller. Αρχικά ελέγχεται ότι ο χρήστης είναι συνδεδεμένος και στη συνέχεια ανακτάται το αρχείο που έχει αποσταλεί από τη φόρμα. Πριν προχωρήσει η αποθήκευση, το backend ελέγχει αν το αρχείο είναι έγκυρο, αν το μέγεθός του βρίσκεται μέσα στα επιτρεπόμενα όρια και αν ο τύπος του αντιστοιχεί σε εικόνα. Το μέγιστο επιτρεπόμενο μέγεθος έχει οριστεί στα 5 MB.

```

230 0 references | 0 overrides
231 public function create()
232 {
233     $this->requireLogin();
234
235     try {
236         $memberId = $this->resolveOwnedMemberId($this->request->getPost('member_id'));
237     } catch (\InvalidArgumentException $e) {
238         return redirect()->back()->with('errors', [$e->getMessage()])->withInput();
239     }
240
241     $uploadedFile = $this->request->getFile('signature_file');
242
243     if ($uploadedFile === null || !$uploadedFile->isValid()) {
244         return redirect()->back()
245             ->with('errors', ['Please upload a valid signature image.'])
246             ->withInput();
247     }
248
249     $sizeError = $this->getUploadSizeError($uploadedFile);
250     if ($sizeError !== null) {
251         return redirect()->back()
252             ->with('errors', [$sizeError])
253             ->withInput();
254     }
255
256     $mimeType = $uploadedFile->getClientMimeType();
257     if ($mimeType === null || !str_starts_with($mimeType, 'image/')) {
258         return redirect()->back()
259             ->with('errors', ['Only image files are allowed.'])
260             ->withInput();
261     }

```

Σχήμα 5.4.3.1: Τμήμα της μεθόδου create() για τον έλεγχο του αρχείου μιας νέας υπογραφής.

Μετά την επιτυχή ολοκλήρωση των ελέγχων, δημιουργείται ένα τυχαίο όνομα για το αρχείο και η εικόνα μεταφέρεται στον κατάλογο αποθήκευσης των υπογραφών. Η χρήση τυχαίου ονόματος αποτρέπει πιθανές συγκρούσεις μεταξύ αρχείων που έχουν το ίδιο αρχικό όνομα. Παράλληλα δημιουργείται η αντίστοιχη εγγραφή στη βάση δεδομένων, στην οποία αποθηκεύονται στοιχεία όπως ο χρήστης στον οποίο ανήκει η υπογραφή, το μέλος εφόσον υπάρχει, ο τίτλος, η περιγραφή, το αρχικό όνομα του αρχείου και η εσωτερική διαδρομή αποθήκευσής του. Τα πεδία που σχετίζονται με την ανάλυση αρχικοποιούνται σε κατάσταση pending, καθώς η αποθήκευση της υπογραφής και η αξιολόγησή της αποτελούν ξεχωριστές διαδικασίες.

```

269     $newName = $uploadedFile->getRandomName();
270     $uploadDirectory = WRITEPATH . 'uploads/signatures';
271
272     if (!is_dir($uploadDirectory)) {
273         mkdir($uploadDirectory, 0775, true);
274     }
275
276     $uploadedFile->move($uploadDirectory, $newName);
277
278     $signature = new Signature([
279         'user_id' => $this->currentUserId(),
280         'member_id' => $memberId,
281         'title' => $this->request->getPost('title'),
282         'content' => $this->request->getPost('content'),
283         'file_name' => $uploadedFile->getClientName(),
284         'file_path' => 'uploads/signatures/' . $newName,
285         'is_genuine' => $isGenuine,
286         'analysis_status' => 'pending',
287         'analysis_result' => null,
288         'analysis_score' => null,
289         'ai_analysis_status' => 'pending',
290         'ai_analysis_result' => null,
291         'ai_analysis_score' => null,
292     ]);
293
294     $id = $this->model->insert($signature);
295     if ($id === false){
296         return redirect()->back()
297             ->with('errors', $this->model->errors())
298             ->withInput();
299     }
300     return redirect()->to("signatures/$id")
301         ->with("message", "Signature saved.");
302 }
303

```

Σχήμα 5.4.3.2: Αποθήκευση εικόνας και δημιουργία της εγγραφής Signature στη μέθοδο create()

5.4.4 Διαχείριση γνήσιων υπογραφών αναφοράς

Οι γνήσιες υπογραφές χρησιμοποιούνται ως σύνολο αναφοράς κατά την αξιολόγηση μιας νέας υπογραφής. Για κάθε χρήστη ή πρόσθετο μέλος μπορούν να αποθηκευτούν έως δέκα γνήσιες υπογραφές. Πριν από την εισαγωγή νέων δειγμάτων, υπολογίζεται πόσες θέσεις παραμένουν διαθέσιμες.

```

153     $existingGenuineCount = $this->model->countGenuineSignatures($this->currentUserId(), $memberId);
154     $remainingSlots = self::MAX_GENUINE_SIGNATURES - $existingGenuineCount;
155     if ($remainingSlots <= 0) {
156         return redirect()->back()
157             ->with('errors', ['This subject already has the maximum of ' . self::MAX_GENUINE_SIGNATURES . ' genuine reference signatures.'])
158             ->withInput();
159     }
160
161     if (count($fileList) > $remainingSlots) {
162         $errorMessagees[] = 'Only ' . $remainingSlots . ' more genuine reference signature(s) can be uploaded for this subject.';
163         $fileList = array_slice($fileList, 0, $remainingSlots);
164     }

```

Σχήμα 5.4.4: Έλεγχος του μέγιστου αριθμού γνήσιων υπογραφών ανά πρόσωπο.

Ο διαχωρισμός των γνήσια signatures πραγματοποιείται ταυτόχρονα με βάση τον χρήστη και το μέλος. Έτσι, οι υπογραφές ενός πρόσθετου μέλους δεν μπορούν να χρησιμοποιηθούν ως δείγματα αναφοράς για διαφορετικό πρόσωπο του ίδιου λογαριασμού.

5.4.5 Έλεγχοι πριν από την ανάλυση

Η κύρια διαδικασία αξιολόγησης ξεκινά από τη μέθοδο `analyze()`. Πριν εκτελεστεί το Python script, πραγματοποιούνται ορισμένοι βασικοί έλεγχοι. Αρχικά επιβεβαιώνεται ότι ο χρήστης είναι συνδεδεμένος και έχει δικαίωμα πρόσβασης στη συγκεκριμένη υπογραφή. Επίσης, μία υπογραφή που έχει ήδη δηλωθεί ως γνήσιο δείγμα δεν επιτρέπεται να χρησιμοποιηθεί ταυτόχρονα ως εξεταζόμενη υπογραφή.

Στη συνέχεια υπολογίζεται ο αριθμός των γνήσιων υπογραφών του συγκεκριμένου προσώπου. Για να είναι διαθέσιμη η ανάλυση απαιτούνται τουλάχιστον πέντε γνήσιες υπογραφές αναφοράς. Τέλος, ελέγχεται ότι το αρχείο της εξεταζόμενης υπογραφής εξακολουθεί να υπάρχει στο filesystem.

```
422 public function analyze($id)
423 {
424     $this->requireLogin();
425     $signature = $this->getSignatureOr404($id);
426     $this->assertCanAccessSignature($signature);
427
428     if ((bool) $signature->is_genuine) {
429         return redirect()->back()->with('message', 'Genuine signatures cannot be analyzed as test signatures.');
```

```
430     }
431
432     $memberId = $this->normalizeMemberId($signature->member_id ?? null);
433     $genuineCount = $this->model->countGenuineSignatures((int) $signature->user_id, $memberId);
434     if ($genuineCount < 5) {
435         return redirect()->back()
436             ->with('message', "You must mark at least 5 genuine signatures before running analysis. You have $genuineCount/5.");
437     }
438
439     $absolutePath = WRITEPATH . $signature->file_path;
440
441     if (!is_file($absolutePath)) {
442         return redirect()->back()->with('message', 'File not found for this signature.');
```

```
443     }
444
445     // Call Python analysis script
446     $analysis = $this->runPythonAnalysis((int) $signature->user_id, $memberId, $absolutePath);
447
448     $this->model->update($id, [
449         'analysis_status' => $analysis['status'],
450         'analysis_result' => $analysis['result'],
451         'analysis_score' => $analysis['score'],
452     ]);
453
454     if ($analysis['status'] !== 'completed') {
455         $errorMessage = $analysis['message'] ?? 'Python analysis failed.';
456         return redirect()->to("signatures/$id")
457             ->with('message', "Analysis failed: " . $errorMessage);
458     }
459
460     return redirect()->to("signatures/$id")
461         ->with('message', 'Analysis finished.');
```

```
462 }
463 }
```

Σχήμα 5.4.5: Έλεγχοι που προηγούνται της ανάλυσης μιας υπογραφής.

5.4.6 Επικοινωνία PHP και Python

Μετά την ολοκλήρωση των απαραίτητων ελέγχων, το backend προετοιμάζει την εκτέλεση του Python script που είναι υπεύθυνο για την ανάλυση της υπογραφής. Αρχικά καθορίζεται η διαδρομή του αρχείου `signature_detection.py`, καθώς και η διαδρομή του Python executable που βρίσκεται στο virtual environment του project. Πριν συνεχιστεί η διαδικασία, ελέγχεται ότι το εκτελέσιμο αρχείο της Python υπάρχει πράγματι στο αναμενόμενο σημείο.

Στη συνέχεια δημιουργείται η εντολή που θα εκτελεστεί από το PHP backend. Ως παράμετροι περνούν το αναγνωριστικό του χρήστη, η διαδρομή της εξεταζόμενης υπογραφής, ο προσωρινός φάκελος που περιέχει τις γνήσιες υπογραφές αναφοράς, καθώς και οι δύο βαρύτητες που χρησιμοποιούνται στον τελικό υπολογισμό του βαθμού. Οι παράμετροι περνούν μέσω της `escapeshellarg()`, ώστε να αντιμετωπίζονται σωστά διαδρομές που περιέχουν κενά ή ειδικούς χαρακτήρες.

```

656 // build python command
657 $pythonScript = rtrim(ROOTPATH, DIRECTORY_SEPARATOR)
658 . DIRECTORY_SEPARATOR . 'SignatureDetection'
659 . DIRECTORY_SEPARATOR . 'signature_detection.py';
660
661 $pythonExecutable = rtrim(ROOTPATH, DIRECTORY_SEPARATOR)
662 . DIRECTORY_SEPARATOR . 'SignatureDetection'
663 . DIRECTORY_SEPARATOR . 'env' . DIRECTORY_SEPARATOR
664 . 'Scripts' . DIRECTORY_SEPARATOR . 'python.exe';
665
666 if (!is_file($pythonExecutable)) {
667     $this->deleteDirectory($tempDir);
668     return [
669         'status' => 'error',
670         'result' => null,
671         'score' => null,
672         'message' => 'Python executable not found at: ' . $pythonExecutable,
673     ];
674 }
675
676 $command = escapeshellarg($pythonExecutable)
677 . ' ' . escapeshellarg($pythonScript)
678 . ' ' . escapeshellarg((string)$userId)
679 . ' ' . escapeshellarg($testSignaturePath)
680 . ' ' . escapeshellarg($tempDir)
681 . ' ' . escapeshellarg((string) round($probWeight, 4))
682 . ' ' . escapeshellarg((string) round($similarityWeight, 4));
683
684 $descriptorspec = [
685     0 => ['pipe', 'r'],
686     1 => ['pipe', 'w'],
687     2 => ['pipe', 'w'],
688 ];
689 $pythonDir = dirname($pythonExecutable);
690 $env = getenv();
691 if (!is_array($env)) {
692     $env = [];
693 }
694
695 $existingPath = (string) ($env['PATH'] ?? getenv('PATH') ?: '');
696 $env['PATH'] = $pythonDir . PATH_SEPARATOR . $existingPath;
697 $env['PYTHONIOENCODING'] = 'utf-8';

```

Σχήμα 5.4.6.1: Δημιουργία εντολής εκτέλεσης και μεταβίβαση των παραμέτρων στο Python script ανάλυσης.

Πριν από την εκτέλεση της διεργασίας διαμορφώνεται επίσης το περιβάλλον στο οποίο θα εκτελεστεί η Python. Ορίζονται τα απαραίτητα pipes για την είσοδο, την κανονική έξοδο και την έξοδο σφαλμάτων, ενώ ενημερώνεται η μεταβλητή PATH ώστε να περιλαμβάνει τον κατάλογο του Python environment. Παράλληλα, ορίζεται η κωδικοποίηση εξόδου σε UTF-8 και αφαιρούνται τυχόν μεταβλητές PYTHONHOME και PYTHONPATH που θα μπορούσαν να επηρεάσουν την εκτέλεση με διαφορετικό Python environment.

Τέλος, η διεργασία ξεκινά μέσω της `proc_open()`. Αν η δημιουργία της διεργασίας αποτύχει, ο προσωρινός φάκελος διαγράφεται και επιστρέφεται κατάλληλο μήνυμα σφάλματος. Με αυτόν τον τρόπο το PHP backend μπορεί να εκτελέσει το Python υποσύστημα ως ξεχωριστή διεργασία και να συνεχίσει στη συνέχεια με την ανάγνωση του αποτελέσματος της ανάλυσης.

```

699     unset($env['PYTHONHOME'], $env['PYTHONPATH']);
700
701     if (!isset($env['SystemRoot']) && getenv('SystemRoot') !== false) {
702         $env['SystemRoot'] = (string) getenv('SystemRoot');
703     }
704     if (!isset($env['WINDIR']) && getenv('WINDIR') !== false) {
705         $env['WINDIR'] = (string) getenv('WINDIR');
706     }
707
708     $process = proc_open($command, $descriptorspec, $pipes, ROOTPATH, $env);
709     if (!$process) {
710         $this->deleteDirectory($tempDir);
711         return [
712             'status' => 'error',
713             'result' => null,
714             'score' => null,
715             'message' => 'Failed to start Python process.',
716         ];
717     }

```

Σχήμα 5.4.6.2: Ρύθμιση του περιβάλλοντος και εκτέλεση του Python script μέσω της `proc_open()`.

5.4.7 Εξαγωγή χαρακτηριστικών και ταξινόμηση

Στο Python υποσύστημα χρησιμοποιείται το DenseNet169 για την εξαγωγή χαρακτηριστικών από τις εικόνες των υπογραφών. Το μοντέλο φορτώνεται με προεκπαιδευμένα βάρη του ImageNet και με `include_top=False`, καθώς δεν χρησιμοποιείται το αρχικό επίπεδο ταξινόμησης του δικτύου. Με αυτόν τον τρόπο, το DenseNet169 αξιοποιείται ως εξαγωγέας χαρακτηριστικών και η έξοδός του χρησιμοποιείται για την παραγωγή ενός διανύσματος χαρακτηριστικών για κάθε εικόνα.

Στη συνέχεια φορτώνεται ο εκπαιδευμένος Naive Bayes ταξινομητής από αρχείο `.pkl`. Ο συγκεκριμένος ταξινομητής χρησιμοποιεί τα χαρακτηριστικά που εξάγονται από το DenseNet169 για να υπολογίσει την πιθανότητα η εξεταζόμενη υπογραφή να ανήκει στην κατηγορία των γνήσιων υπογραφών. Ο συνδυασμός DenseNet169 και Naive Bayes επιλέχθηκε με βάση τα αποτελέσματα της πειραματικής μελέτης που παρουσιάστηκε στο Κεφάλαιο 3.

```

23 try:
24     from tensorflow.keras.applications import DenseNet169
25     from tensorflow.keras.applications.densenet import preprocess_input
26     from tensorflow.keras.models import Model
27 except Exception as exc:
28     TF_IMPORT_ERROR = str(exc)
29     DenseNet169 = None
30     preprocess_input = None
31     Model = None
32
33 model = None
34 if TF_IMPORT_ERROR is None:
35     try:
36         base_model = DenseNet169(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
37         model = Model(inputs=base_model.input, outputs=base_model.output)
38     except Exception as exc:
39         MODEL_INIT_ERROR = str(exc)
40
41 def preprocess_image(filepath):
42     image = cv2.imread(filepath)
43     if image is None:
44         raise ValueError(f"Image not found or could not be loaded: {filepath}")
45     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
46     image = cv2.resize(image, (224, 224))
47     image = preprocess_input(image)
48     return np.expand_dims(image, axis=0)
49
50 def extract_features(filepath):
51     if model is None:
52         reason = MODEL_INIT_ERROR or TF_IMPORT_ERROR or 'Unknown TensorFlow initialization failure'
53         raise RuntimeError(f"TensorFlow model is unavailable: {reason}")
54
55     image = preprocess_image(filepath)
56     features = model.predict(image)
57     return features.flatten()
58
59 model_path = os.path.join(script_dir, 'models', 'DenseNet169_Naive Bayes_model.pkl')
60 classifier = None
61 try:
62     with open(model_path, 'rb') as file:
63         classifier = pickle.load(file)
64 except Exception as exc:
65     CLASSIFIER_LOAD_ERROR = str(exc)

```

Σχήμα 5.4.7.1: Φόρτωση του DenseNet169 ως εξαγωγή χαρακτηριστικών και του εκπαιδευμένου ταξινομητή Naive Bayes.

Πριν από την εξαγωγή των χαρακτηριστικών, κάθε εικόνα υποβάλλεται σε βασική προεπεξεργασία. Η εικόνα φορτώνεται μέσω OpenCV, μετατρέπεται από BGR σε RGB και αλλάζει διαστάσεις σε 224×224 pixels, ώστε να είναι συμβατή με την είσοδο του DenseNet169. Στη συνέχεια εφαρμόζεται η αντίστοιχη συνάρτηση `preprocess_input()` και προστίθεται μία επιπλέον διάσταση ώστε η εικόνα να μπορεί να δοθεί στο μοντέλο.

```

41 def preprocess_image(filepath):
42     image = cv2.imread(filepath)
43     if image is None:
44         raise ValueError(f"Image not found or could not be loaded: {filepath}")
45     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
46     image = cv2.resize(image, (224, 224))
47     image = preprocess_input(image)
48     return np.expand_dims(image, axis=0)

```

Σχήμα 5.4.7.2: Προεπεξεργασία της εικόνας

Η εξαγωγή των χαρακτηριστικών πραγματοποιείται μέσω της συνάρτησης `extract_features()`. Η προεπεξεργασμένη εικόνα περνά από το `DenseNet169` και η έξοδος του μοντέλου μετατρέπεται σε μονοδιάστατο διάνυσμα μέσω της `flatten()`. Το διάνυσμα αυτό αποτελεί την αναπαράσταση της υπογραφής και χρησιμοποιείται στη συνέχεια τόσο από τον Naive Bayes ταξινομητή όσο και κατά τον υπολογισμό της συνημιτονικής ομοιότητας.

```
50 def extract_features(filepath):
51     if model is None:
52         reason = MODEL_INIT_ERROR or TF_IMPORT_ERROR or 'Unknown TensorFlow initialization failure'
53         raise RuntimeError(f"TensorFlow model is unavailable: {reason}")
54
55     image = preprocess_image(filepath)
56     features = model.predict(image)
57     return features.flatten()
```

Σχήμα 5.4.7.3: Εξαγωγή του διανύσματος χαρακτηριστικών μέσω DenseNet169

5.4.8 Υπολογισμός ομοιότητας και τελικού βαθμού

Μετά την εξαγωγή των χαρακτηριστικών της εξεταζόμενης υπογραφής, το διάνυσμά της συγκρίνεται με τα αντίστοιχα διανύσματα των γνήσιων υπογραφών αναφοράς. Για τη σύγκριση χρησιμοποιείται η συνημιτονική ομοιότητα, η οποία υπολογίζει τον βαθμό ομοιότητας μεταξύ δύο διανυσμάτων χαρακτηριστικών.

Στην υλοποίηση, η συνημιτονική ομοιότητα υπολογίζεται ξεχωριστά μεταξύ της εξεταζόμενης υπογραφής και κάθε γνήσιας υπογραφής του συνόλου αναφοράς. Οι επιμέρους τιμές αποθηκεύονται σε λίστα και στη συνέχεια υπολογίζεται ο μέσος όρος τους. Με αυτόν τον τρόπο προκύπτει μία συνολική τιμή που εκφράζει πόσο κοντά βρίσκεται η εξεταζόμενη υπογραφή στις γνωστές γνήσιες υπογραφές του συγκεκριμένου προσώπου.

```
77 # Extract features from the test signature
78 test_features = extract_features(filepath)
79
80 # Calculate similarity with each genuine signature
81 similarities = []
82 for genuine_feature in genuine_features:
83     sim = cosine_similarity([genuine_feature], [test_features])[0][0] # Cosine similarity
84     similarities.append(sim)
85
86 # Average the similarities
87 avg_similarity = np.mean(similarities)
```

Σχήμα 5.4.8.1: Υπολογισμός της συνημιτονικής ομοιότητας μεταξύ της εξεταζόμενης υπογραφής και των γνήσιων υπογραφών αναφοράς

Στη συνέχεια, το ίδιο διάνυσμα χαρακτηριστικών δίνεται στον εκπαιδευμένο ταξινομητή Naive Bayes. Μέσω της `predict_proba()` λαμβάνεται η πιθανότητα η υπογραφή να ανήκει στην κατηγορία των γνήσιων υπογραφών. Η πιθανότητα αυτή συνδυάζεται με τη μέση συνημιτονική ομοιότητα, ώστε να προκύψει ο τελικός βαθμός αξιολόγησης.

```
89 # Combine the classifier's probability with the similarity score
90 test_features = test_features.reshape(1, -1) # Reshape for prediction
91 prob = classifier.predict_proba(test_features)[0][1] # Probability of being genuine
92 combined_score = prob_weight * prob + similarity_weight * avg_similarity # Weighted combination
93
94 # Final decision based on the combined score
95 return combined_score
```

Σχήμα 5.4.8.2: Συνδυασμός της πιθανότητας του ταξινομητή με τη μέση συνημιτονική ομοιότητα

Με βάση τα αποτελέσματα της πειραματικής μελέτης του Κεφαλαίου 3, οι προεπιλεγμένες βαρύτητες έχουν οριστεί σε 35% για την πιθανότητα του μοντέλου και 65% για τη μέση συνημιτονική ομοιότητα. Επομένως, με τις προεπιλεγμένες ρυθμίσεις ο υπολογισμός μπορεί να εκφραστεί ως:

$$Score = 0.35 \cdot Pmodel + 0.65 \cdot Scosine$$

όπου $Pmodel$ είναι η πιθανότητα γνησιότητας που επιστρέφει ο Naive Bayes και $Scosine$ η μέση συνημιτονική ομοιότητα με τις γνήσιες υπογραφές αναφοράς. Οι συγκεκριμένες βαρύτητες μπορούν να μεταβληθούν από τον χρήστη μέσω των ρυθμίσεων της εφαρμογής.

Η τελική απόφαση λαμβάνεται με κατώφλι 0,5. Αν ο συνδυαστικός βαθμός είναι μεγαλύτερος ή ίσος του 0,5, η υπογραφή χαρακτηρίζεται ως γνήσια, ενώ σε διαφορετική περίπτωση χαρακτηρίζεται ως πλαστή.

```
177     score = test_signature(test_signature_path, genuine_features, prob_weight, similarity_weight)
178     result = 'genuine' if score >= 0.5 else 'forged'
179
180     return {
181         'status': 'completed',
182         'score': float(score),
183         'result': result
184     }
```

Σχήμα 5.4.8.3: Παραγωγή του τελικού αποτελέσματος

5.4.9 Αποθήκευση του αποτελέσματος της ανάλυσης

Μετά την ολοκλήρωση της διαδικασίας στην Python, το αποτέλεσμα επιστρέφεται στον Signatures controller. Η μέθοδος `analyze()` ενημερώνει την αντίστοιχη εγγραφή της υπογραφής με την κατάσταση της ανάλυσης, το αποτέλεσμα και τον τελικό βαθμό. Με αυτόν τον τρόπο, το αποτέλεσμα παραμένει αποθηκευμένο στη βάση δεδομένων και μπορεί να εμφανιστεί ξανά χωρίς να απαιτείται νέα εκτέλεση της ανάλυσης.

Συγκεκριμένα, το πεδίο `analysis_status` δηλώνει αν η διαδικασία ολοκληρώθηκε επιτυχώς, το `analysis_result` περιέχει την τελική ένδειξη γνήσια ή πλαστή, ενώ στο `analysis_score` αποθηκεύεται η αριθμητική τιμή που επέστρεψε το Python script. Σε περίπτωση αποτυχίας, ο χρήστης μεταφέρεται ξανά στη σελίδα της υπογραφής και εμφανίζεται το αντίστοιχο μήνυμα σφάλματος.

```
445     $analysis = $this->runPythonAnalysis((int) $signature->user_id, $memberId, $absolutePath);
446
447     $this->model->update($id, [
448         'analysis_status' => $analysis['status'],
449         'analysis_result' => $analysis['result'],
450         'analysis_score' => $analysis['score'],
451     ]);
452
453     if ($analysis['status'] !== 'completed') {
454         $errorMessage = $analysis['message'] ?? 'Python analysis failed.';
455         return redirect()->to("signatures/$id")
456             ->with('message', 'Analysis failed: ' . $errorMessage);
457     }
458
459     return redirect()->to("signatures/$id")
460         ->with('message', 'Analysis finished.');
```

Σχήμα 5.4.9: Αποθήκευση του αποτελέσματος της ανάλυσης στη βάση δεδομένων μέσω της μεθόδου `analyze()`.

5.4.10 Εναλλακτική αξιολόγηση μέσω OpenAI

Εκτός από τον βασικό μηχανισμό ανάλυσης, η Signalyze παρέχει τη δυνατότητα ξεχωριστής αξιολόγησης μιας υπογραφής μέσω του OpenAI API. Η λειτουργία αυτή υλοποιείται στη μέθοδο `analyzeAi()` του Signatures controller και λειτουργεί ανεξάρτητα από την ανάλυση που βασίζεται στο DenseNet169, τον Naive Bayes και την συννημιτονική ομοιότητα.

Πριν πραγματοποιηθεί οποιοδήποτε αίτημα προς το API, εφαρμόζονται οι βασικοί έλεγχοι που χρησιμοποιούνται και στην κύρια ανάλυση. Ο χρήστης πρέπει να είναι συνδεδεμένος, η υπογραφή πρέπει να ανήκει στον λογαριασμό του και δεν πρέπει να είναι ήδη χαρακτηρισμένη ως γνήσια υπογραφή αναφοράς. Επιπλέον, απαιτείται η ύπαρξη τουλάχιστον πέντε γνήσιων υπογραφών για το ίδιο πρόσωπο.

Η εφαρμογή ελέγχει επίσης αν ο χρήστης έχει καταχωρίσει OpenAI API key στο προφίλ του. Το αποθηκευμένο key αποκρυπτογραφείται πριν χρησιμοποιηθεί για την κλήση της υπηρεσίας. Αν δεν υπάρχει key ή δεν είναι δυνατή η αποκρυπτογράφησή του, η διαδικασία δεν συνεχίζεται.

```
0 references | 0 overrides
463 public function analyzeAi($id)
464 {
465     $this->requireLogin();
466     $signature = $this->getSignatureOr404($id);
467     $this->assertCanAccessSignature($signature);
468
469     if ((bool) $signature->is_genuine) {
470         return redirect()->back()->with('message', 'Genuine signatures cannot be analyzed as test signatures.');
```

```
471     }
472
473     $memberId = $this->normalizeMemberId($signature->member_id ?? null);
474     $genuineCount = $this->model->countGenuineSignatures((int) $signature->user_id, $memberId);
475     if ($genuineCount < 5) {
476         return redirect()->back()
477             ->with('message', "You must mark at least 5 genuine signatures before running AI analysis. You have $genuineCount/5.");
478     }
479
480     $absolutePath = WRITEPATH . $signature->file_path;
481     if (!is_file($absolutePath)) {
482         return redirect()->back()->with('message', 'File not found for this signature.');
```

```
483     }
484
485     $owner = $this->userModel->find((int) $signature->user_id);
486     $storedOpenAiKey = trim((string) ($owner->openai_api_key ?? ''));
487     if ($storedOpenAiKey === '') {
488         return redirect()->to('/profile/edit')->with('message', 'Please add your OpenAI API key in Profile before using AI test.');
```

```
489     }
490
491     try {
492         $openAiApiKey = OpenAiKeyCipher::decrypt($storedOpenAiKey);
493     } catch (\RuntimeException $e) {
494         return redirect()->to('/profile/edit')->with('message', 'Saved OpenAI API key could not be decrypted. Please save it again.');
```

```
495     }
496
497     if (!OpenAiKeyCipher::isEncrypted($storedOpenAiKey)) {
498         $owner->openai_api_key = OpenAiKeyCipher::encrypt($openAiApiKey);
499         $this->userModel->save($owner);
500     }
501
502     $analysis = $this->runOpenAiAnalysis((int) $signature->user_id, $memberId, $absolutePath, $openAiApiKey);
503
504     $this->model->update($id, [
505         'ai_analysis_status' => $analysis['status'],
506         'ai_analysis_result' => $analysis['result'],
507         'ai_analysis_score' => $analysis['score'],
508     ]);
509
510     if ($analysis['status'] !== 'completed') {
511         $errorMessage = $analysis['message'] ?? 'AI analysis failed.';
512         return redirect()->to("signatures/$id")
513             ->with('message', "AI analysis failed: " . $errorMessage);
514     }
515
516     return redirect()->to("signatures/$id")
517         ->with('message', 'AI analysis finished.');
```

```
518 }
```

Σχήμα 5.4.10.1: Έλεγχος των προϋποθέσεων πριν από την εναλλακτική αξιολόγηση μέσω OpenAI.

Η μέθοδος runOpenAiAnalysis() επιλέγει τις γνήσιες υπογραφές που αντιστοιχούν στον ίδιο χρήστη ή μέλος. Για την αξιολόγηση χρησιμοποιούνται οι πέντε πρώτες γνήσιες υπογραφές και η εξεταζόμενη υπογραφή. Οι εικόνες μετατρέπονται σε data URLs και προστίθενται στο περιεχόμενο του αιτήματος.

```
848 $images = [];
849 foreach (array_slice($userGenuineSignatures, 0, 5) as $sig) {
850     $path = WRITEPATH . $sig->file_path;
851     if (is_file($path)) {
852         $images[] = [
853             'kind' => 'genuine',
854             'data_url' => $this->buildImageDataUrl($path),
855         ];
856     }
857 }
858
859 if (count($images) < 5) {
860     return [
861         'status' => 'error',
862         'result' => null,
863         'score' => null,
864         'message' => 'Unable to load 5 genuine signature images for AI analysis.',
865     ];
866 }
867
868 $images[] = [
869     'kind' => 'test',
870     'data_url' => $this->buildImageDataUrl($testSignaturePath),
871 ];
```

Σχήμα 5.4.10.2: Προσθήκη των εικόνων στο αίτημα

Στη συνέχεια δημιουργείται το prompt που περιγράφει στο μοντέλο τη διαδικασία που πρέπει να ακολουθήσει. Το μοντέλο καλείται να συγκρίνει τις πέντε γνήσιες υπογραφές αναφοράς με την εξεταζόμενη υπογραφή και να επιστρέφει αποτέλεσμα σε συγκεκριμένη μορφή JSON.

```
873 $content = [
874     [
875         'type' => 'input_text',
876         'text' => 'You are a signature verification assistant. Compare five known ' .
877             'genuine signatures against one test signature. Return strict JSON only ' .
878             'with keys: result and certainty_score. result must be either genuine or forged. ' .
879             'certainty_score must be a number from 0 to 100 where higher means more certain.',
880     ],
881 ];
882
883 foreach ($images as $index => $image) {
884     $label = $image['kind'] === 'genuine'
885         ? 'Reference genuine signature #' . ($index + 1)
886         : 'Test signature to classify';
887
888     $content[] = [
889         'type' => 'input_text',
890         'text' => $label,
891     ];
892     $content[] = [
893         'type' => 'input_image',
894         'image_url' => $image['data_url'],
895     ];
896 }
```

Σχήμα 5.4.10.3: Δημιουργία του prompt

5.4.11 Αποστολή και έλεγχος της απάντησης του OpenAI API

Μετά τη δημιουργία του περιεχομένου, σχηματίζεται το request προς το OpenAI Responses API. Στην τρέχουσα υλοποίηση χρησιμοποιείται το μοντέλο gpt-4.1-mini, ενώ το API key μεταφέρεται μέσω του Authorization header. Το request περιλαμβάνει τόσο τις οδηγίες όσο και τις εικόνες που πρόκειται να συγκριθούν.

```
898     $payload = [  
899         'model' => 'gpt-4.1-mini',  
900         'input' => [  
901             [  
902                 'role' => 'user',  
903                 'content' => $content,  
904             ],  
905         ],  
906     ];  
907  
908     $ch = curl_init('https://api.openai.com/v1/responses');  
909     curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
910     curl_setopt($ch, CURLOPT_POST, true);  
911     curl_setopt($ch, CURLOPT_HTTPHEADER, [  
912         'Authorization: Bearer ' . $apiKey,  
913         'Content-Type: application/json',  
914     ]);  
915     curl_setopt($ch, CURLOPT_TIMEOUT, 120);  
916     curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));  
917  
918     $raw = curl_exec($ch);  
919     $httpCode = (int) curl_getinfo($ch, CURLINFO_HTTP_CODE);  
920     $curlError = curl_error($ch);  
921     curl_close($ch);
```

Σχήμα 5.4.11.1: Δημιουργία και αποστολή του request προς το OpenAI Responses API.

Η απάντηση που επιστρέφει το API δεν αποθηκεύεται απευθείας. Αρχικά ελέγχεται αν μπορεί να μετατραπεί σε JSON και αν ο HTTP status code υποδηλώνει επιτυχημένη κλήση. Στη συνέχεια εξάγεται το JSON αντικείμενο που έχει δημιουργήσει το μοντέλο και ελέγχεται ότι το πεδίο result περιέχει αποκλειστικά μία από τις τιμές γνήσια ή πλαστή.

Το certainty_score μετατρέπεται σε αριθμητική τιμή και περιορίζεται στο διάστημα 0 έως 100. Η συγκεκριμένη τιμή δεν είναι ο ίδιος βαθμός με εκείνον της κύριας ανάλυσης. Στην περίπτωση του OpenAI πρόκειται για τον βαθμό βεβαιότητας που επιστρέφει το μοντέλο και όχι για αποτέλεσμα του συνδυασμού της πιθανότητας του ταξινομητή με τη συνημιτονική ομοιότητα.

```

963     $result = strtolower(trim((string) ($json['result'] ?? '')));
964     if (!in_array($result, ['genuine', 'forged'], true)) {
965         return [
966             'status' => 'error',
967             'result' => null,
968             'score' => null,
969             'message' => 'AI response returned an invalid result label.',
970         ];
971     }
972
973     $certaintyScore = (float) ($json['certainty_score'] ?? 0);
974     if ($certaintyScore < 0) {
975         $certaintyScore = 0;
976     }
977     if ($certaintyScore > 100) {
978         $certaintyScore = 100;
979     }
980
981     return [
982         'status' => 'completed',
983         'result' => $result,
984         'score' => round($certaintyScore, 2),
985         'message' => null,
986     ];

```

Σχήμα 5.4.11.2: Έλεγχος και επικύρωση του αποτελέσματος που επιστρέφει το OpenAI API.

Τέλος, η μέθοδος analyzeAi() αποθηκεύει το αποτέλεσμα σε ξεχωριστά πεδία του πίνακα signature, ώστε να μην αντικαθίσταται η βασική ανάλυση της εφαρμογής.

```

504     $this->model->update($id, [
505         'ai_analysis_status' => $analysis['status'],
506         'ai_analysis_result' => $analysis['result'],
507         'ai_analysis_score' => $analysis['score'],
508     ]);

```

Σχήμα 5.4.11.3: Αποθήκευση των αποτελεσμάτων

Με αυτόν τον διαχωρισμό, τα analysis_result και analysis_score αντιστοιχούν στη βασική διαδικασία που συνδυάζει το DenseNet169, τον Naive Bayes και τη συνημιτονική ομοιότητα, ενώ τα ai_analysis_result και ai_analysis_score αφορούν αποκλειστικά την εναλλακτική αξιολόγηση μέσω OpenAI.

5.4.12 Συνολική ροή του Backend

Η υλοποίηση του backend της Signalyze συνδέει τις επιμέρους λειτουργίες της εφαρμογής σε μία ενιαία διαδικασία. Αρχικά, κάθε αίτημα περνά από τους απαραίτητους ελέγχους αυθεντικοποίησης και πρόσβασης. Κατά την εισαγωγή μιας υπογραφής ελέγχεται το αρχείο, αποθηκεύεται στο filesystem και δημιουργείται η αντίστοιχη εγγραφή στη βάση δεδομένων.

Όταν ζητείται η ανάλυση μιας υπογραφής, το backend επιβεβαιώνει ότι υπάρχουν τουλάχιστον πέντε γνήσια δείγματα για το ίδιο πρόσωπο. Στη συνέχεια δημιουργείται το κατάλληλο σύνολο αναφοράς και καλείται το Python script. Το DenseNet169 χρησιμοποιείται για την εξαγωγή χαρακτηριστικών, ενώ ο Naive Bayes παράγει την πιθανότητα γνησιότητας. Παράλληλα, υπολογίζεται η μέση συνημιτονική ομοιότητα της εξεταζόμενης υπογραφής προς τις γνήσιες υπογραφές αναφοράς.

Οι δύο τιμές συνδυάζονται με τις επιλεγμένες βαρύτητες. Ως προεπιλογή χρησιμοποιούνται 35% για την πιθανότητα του ταξινομητή και 65% για τη συνημιτονική ομοιότητα, σύμφωνα με τα αποτελέσματα της πειραματικής μελέτης του Κεφαλαίου 3. Ο συνδυαστικός βαθμός συγκρίνεται με το κατώφλι 0,5 και παράγεται η τελική ένδειξη «γνήσια» ή «πλαστή».

Παράλληλα, η εφαρμογή παρέχει μία ανεξάρτητη εναλλακτική αξιολόγηση μέσω OpenAI, κατά την οποία πέντε γνήσιες υπογραφές και μία εξεταζόμενη υπογραφή αποστέλλονται στο μοντέλο ως εικόνες αναφοράς. Τα αποτελέσματα αυτής της διαδικασίας αποθηκεύονται χωριστά από την κύρια ανάλυση.

Με αυτόν τον τρόπο, το backend λειτουργεί ως το ενδιάμεσο επίπεδο που συνδέει το περιβάλλον χρήστη, τη βάση δεδομένων, το filesystem και τα δύο διαφορετικά υποσυστήματα αξιολόγησης της υπογραφής.

5.5 Υλοποίηση του Frontend

Το frontend της Signalyze υλοποιήθηκε με PHP views του CodeIgniter 4, HTML, CSS, JavaScript και Bootstrap 5.3.3. Τα views λειτουργούν ως το επίπεδο παρουσίασης της εφαρμογής και λαμβάνουν από τους controllers τα δεδομένα που απαιτούνται για τη δημιουργία των επιμέρους σελίδων. Με αυτόν τον τρόπο, στοιχεία όπως οι υπογραφές, τα πρόσθετα μέλη, τα αποτελέσματα των αναλύσεων και οι επιλογές του χρήστη εμφανίζονται δυναμικά ανάλογα με την κατάσταση της εφαρμογής.

Για να διατηρείται κοινή εμφάνιση σε όλες τις σελίδες, τα views βασίζονται σε ένα κεντρικό layout, ενώ το Bootstrap χρησιμοποιείται για τη διάταξη και την προσαρμογή σε διαφορετικά μεγέθη οθόνης. Η JavaScript αξιοποιείται κυρίως στις λειτουργίες επιλογής εικόνων, λήψης φωτογραφίας από κινητή συσκευή, drag-and-drop και τοπικής προεπισκόπησης αρχείων πριν από την αποστολή τους στον server.

Στην παρούσα ενότητα παρουσιάζονται αντιπροσωπευτικά αποσπάσματα της υλοποίησης του frontend. Η αναλυτική παρουσίαση των τελικών οθονών και της χρήσης της εφαρμογής πραγματοποιείται στο επόμενο κεφάλαιο.

5.5.1 Κοινό layout και δομή των views

Οι διαφορετικές σελίδες της εφαρμογής δεν περιλαμβάνουν η καθεμία ξεχωριστά ολόκληρη τη βασική HTML δομή. Αντίθετα, χρησιμοποιείται το κοινό layout `app/Views/layouts/default.php`, το οποίο περιλαμβάνει στοιχεία που επαναλαμβάνονται σε όλη την εφαρμογή, όπως το `<head>`, τη γραμμή πλοήγησης, τις γραμματοσειρές, το Bootstrap και το βασικό container του περιεχομένου.

Τα επιμέρους views συμπληρώνουν συγκεκριμένες sections του layout. Η `renderSection("title")` χρησιμοποιείται για τον τίτλο της σελίδας, ενώ η `renderSection("content")` καθορίζει το σημείο στο οποίο εισάγεται το κύριο περιεχόμενο κάθε view. Έτσι αποφεύγεται η επανάληψη του ίδιου HTML κώδικα και πιθανές αλλαγές στο navigation ή στις κοινές βιβλιοθήκες μπορούν να πραγματοποιούνται κεντρικά.

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5
6      <meta name="viewport"
7          content="width=device-width, initial-scale=1.0">
8
9      <title><?= $this->renderSection("title") ?></title>
10
11      <link
12          href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
13          rel="stylesheet">
14  </head>
15
16  <body>
17
18  <nav class="navbar navbar-expand-lg navbar-light bg-light">
19
20      <div class="container-fluid">
21
22          <a class="navbar-brand" href="/">
23              <span class="brand-dot"></span>
24              Signalyze
25          </a>
26
27      </div>
28
29  </nav>
30
31  <main class="content-shell">
32      <?= $this->renderSection("content") ?>
33  </main>
34
35  </body>
36  </html>

```

Σχήμα 5.5.1.1: Ενδεικτικό παράδειγμα κοινού layout των views και εισαγωγή του δυναμικού περιεχομένου μέσω των sections του CodeIgniter.

Το ίδιο layout προσαρμόζει και τις επιλογές της γραμμής πλοήγησης ανάλογα με το αν υπάρχει συνδεδεμένος χρήστης. Όταν δεν υπάρχει ενεργό session εμφανίζονται οι επιλογές Login και Register, ενώ μετά τη σύνδεση εμφανίζεται το όνομα του χρήστη και menu με τις επιλογές Profile και Logout. Η συγκεκριμένη λογική επιτρέπει τη χρήση του ίδιου navigation σε όλες τις καταστάσεις της εφαρμογής.

```

210  <div class="navbar-auth ms-lg-auto order-lg-2">
211      <?php if (auth()->loggedIn()): ?>
212          <div class="dropdown">
213              <button class="badge text-bg-light border dropdown-toggle" style="border: none; background: none; cursor: pointer;" type="button">
214                  Hello, <?= esc(auth()->user()->first_name) ?>
215              </button>
216              <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="userMenu">
217                  <li><a class="dropdown-item" href="<?= url_to('Profile::show') ?>">Profile</a></li>
218                  <li><hr class="dropdown-divider"></li>
219                  <li><a class="dropdown-item" href="<?= url_to('logout') ?>">Logout</a></li>
220              </ul>
221          </div>
222      <?php else: ?>
223          <a href="<?= url_to('login') ?>" class="btn btn-sm navbar-login-action">Login</a>
224          <a href="<?= url_to('register') ?>" class="btn btn-sm navbar-register-action">Register</a>
225      <?php endif; ?>
226  </div>

```

Σχήμα 5.5.1.2: Δυναμική προσαρμογή της γραμμής πλοήγησης ανάλογα με την κατάσταση σύνδεσης.

5.5.2 Φόρμα εισαγωγής και επεξεργασίας υπογραφής

Η κοινή φόρμα για τη δημιουργία και την επεξεργασία μιας υπογραφής βρίσκεται στο `app/Views/Signatures/form.php`. Το ίδιο view χρησιμοποιείται και στις δύο περιπτώσεις, ενώ η μεταβλητή `$isEdit` καθορίζει αν η φόρμα αφορά νέα εγγραφή ή επεξεργασία υπάρχουσας υπογραφής.

Η φόρμα περιλαμβάνει τον τίτλο και την περιγραφή της υπογραφής, την επιλογή του προσώπου στο οποίο ανήκει, καθώς και το αρχείο εικόνας. Η επιλογή του προσώπου γίνεται μέσω του πεδίου `member_id`. Όταν δεν έχει επιλεγεί κάποιον πρόσθετο μέλος, η υπογραφή θεωρείται ότι ανήκει στον ίδιο τον κάτοχο του λογαριασμού. Τα διαθέσιμα μέλη εμφανίζονται δυναμικά μέσα στο `<select>` από τα δεδομένα που έχουν αποσταλεί στο view από τον controller.

```
1  <?php $isEdit = !empty($signature->id); ?>
2
3  <div class="mb-3">
4      <label for="title" class="form-label fw-semibold">Title</label>
5      <input type="text" id="title" name="title" value="<?= old("title", esc($signature->title)) ?>" class="form-control" required>
6      <div class="invalid-feedback">
7          Please provide a title.
8      </div>
9  </div>
10
11 <div class="mb-3">
12     <label for="content" class="form-label fw-semibold">Content</label>
13     <textarea id="content" name="content" class="form-control" rows="3"><?= old("content", esc($signature->content)) ?></textarea>
14     <div class="invalid-feedback">
15         Please provide content.
16     </div>
17 </div>
18
19 <div class="mb-3">
20     <label for="member_id" class="form-label fw-semibold">Signature Subject</label>
21     <select id="member_id" name="member_id" class="form-select">
22         <?php $chosenMemberId = old('member_id', $selectedMemberId ?? ($signature->member_id ?? '')); ?>
23         <option value="">Your account</option>
24         <?php foreach (($members ?? []) as $member): ?>
25             <option value="<?= esc($member['id']) ?>" <?= (string) $chosenMemberId === (string) $member['id'] ? 'selected' : '' ?>>
26                 <?= esc($member['name']) ?>
27             </option>
28         <?php endforeach; ?>
29     </select>
30     <small class="text-muted d-block mt-1">Choose who this signature belongs to before adding it to the reference set or analyzing it.</small>
31 </div>
```

Σχήμα 5.5.2.1: Πεδία τίτλου, περιγραφής και επιλογής προσώπου στη φόρμα υπογραφής.

Στο ίδιο view περιλαμβάνεται και η επιλογή εικόνας. Το πραγματικό input που αποστέλλεται στον server είναι το `signature_file`, ενώ επιπλέον χρησιμοποιούνται δύο βοηθητικά file inputs για επιλογή υπάρχουσας φωτογραφίας ή λήψη νέας φωτογραφίας από κινητή συσκευή. Η συγκεκριμένη λειτουργία αναλύεται στην επόμενη υποενότητα.

```
33 <div class="mb-3">
34     <label for="signature_file" class="form-label fw-semibold">Signature Image (PNG/JPG, maximum 5 MB)</label>
35     <div class="d-flex flex-wrap gap-2 mb-2">
36         <button type="button" id="choosePhotoBtn" class="btn btn-outline-primary btn-sm">Choose existing photo</button>
37         <button type="button" id="takePhotoBtn" class="btn btn-outline-secondary btn-sm mobile-only-camera">Take a photo</button>
38     </div>
39     <input type="file" id="galleryInput" accept="image/*" style="position: absolute; width: 1px; height: 1px; opacity: 0; pointer-events: none;">
40     <input type="file" id="cameraInput" accept="image/*" capture="environment" style="position: absolute; width: 1px; height: 1px; opacity: 0; pointer-events: none;">
41     <input type="file" id="signature_file" name="signature_file" class="form-control d-none" accept="image/*" <?= $isEdit ? '' : 'required' ?>>
42     <div id="selectedFileFeedback" class="form-text text-success mt-2 d-none"></div>
43 </div>
```

Σχήμα 5.5.2.2: Επιλογή αρχείου

Κατά τη δημιουργία μιας νέας υπογραφής εμφανίζεται επιπλέον η επιλογή `is_genuine`, μέσω της οποίας ο χρήστης μπορεί να δηλώσει ότι η εικόνα αποτελεί γνήσια υπογραφή αναφοράς. Η επιλογή αυτή δεν εμφανίζεται με τον ίδιο τρόπο κατά την επεξεργασία, καθώς η κατάσταση της υπογραφής μπορεί να μεταβληθεί από ξεχωριστή λειτουργία της εφαρμογής.

```

109 <?php if (!$isEdit): ?>
110     <div class="mb-3 form-check">
111         <input type="checkbox" id="is_genuine" name="is_genuine" class="form-check-input" value="1">
112         <label class="form-check-label" for="is_genuine">
113             Mark as Genuine Reference Signature
114         </label>
115         <small class="d-block text-muted mt-1">
116             <?php if (isset($trainingNeeded) && $trainingNeeded): ?>
117                 Mark this as a genuine reference signature (you need 5 total).
118             <?php endif; ?>
119         </small>
120     </div>
121 <?php endif; ?>

```

Σχήμα 5.5.2.3: Δυνατότητα χαρακτηρισμού μιας νέας υπογραφής ως γνήσιας υπογραφής αναφοράς

Όταν η φόρμα χρησιμοποιείται για επεξεργασία, το αρχείο εικόνας δεν είναι υποχρεωτικό, καθώς ο χρήστης μπορεί να αλλάξει μόνο τα υπόλοιπα στοιχεία της εγγραφής. Αν υπάρχει ήδη αποθηκευμένο αρχείο, εμφανίζεται το όνομά του ώστε να είναι σαφές ποια εικόνα είναι συνδεδεμένη με την υπογραφή.

5.5.3 Επιλογή εικόνας από αρχείο ή κάμερα

Για τη διευκόλυνση της χρήσης της εφαρμογής τόσο σε υπολογιστές όσο και σε κινητές συσκευές, η φόρμα υποστηρίζει δύο διαφορετικούς τρόπους εισαγωγής εικόνας: την επιλογή υπάρχουσας φωτογραφίας και τη λήψη νέας φωτογραφίας μέσω της κάμερας.

Στο frontend χρησιμοποιούνται δύο ξεχωριστά file inputs. Το galleryInput χρησιμοποιείται για την επιλογή υπάρχουσας εικόνας, ενώ το cameraInput διαθέτει την ιδιότητα capture="environment", ώστε σε συμβατές κινητές συσκευές να μπορεί να ενεργοποιηθεί η πίσω κάμερα. Τα συγκεκριμένα inputs παραμένουν κρυφά από τον χρήστη και ενεργοποιούνται μέσω των αντίστοιχων κουμπιών Choose existing photo και Take a photo.

Παράλληλα, το signature_file αποτελεί το πραγματικό input της φόρμας που αποστέλλεται τελικά στον server. Με αυτόν τον τρόπο, ανεξάρτητα από το αν η εικόνα επιλεγεί από τη συσκευή ή ληφθεί μέσω της κάμερας, η φόρμα συνεχίζει να χρησιμοποιεί το ίδιο πεδίο για την αποστολή του αρχείου.

5.5.4 Μαζική προσθήκη και προεπισκόπηση γνήσιων υπογραφών

Για την αρχική καταχώριση των γνήσιων δειγμάτων έχει υλοποιηθεί ξεχωριστή σελίδα στο training.php, η οποία επιτρέπει την επιλογή πολλών εικόνων ταυτόχρονα. Εκτός από το κλασικό file picker, υποστηρίζεται και drag-and-drop, ώστε ο χρήστης να μπορεί να μεταφέρει απευθείας τις εικόνες στη σχετική περιοχή της σελίδας.

Οι επιλεγμένες εικόνες διατηρούνται προσωρινά στον πίνακα selectedFiles. Η συνάρτηση handleFiles() φιλτράρει τα αρχεία ώστε να γίνονται δεκτές μόνο εικόνες και ελέγχει αν κάποιο αρχείο έχει ήδη προστεθεί. Για την αναγνώριση διπλότυπων συγκρίνονται το όνομα, το μέγεθος και η ημερομηνία τελευταίας τροποποίησης.

```

223 let selectedFiles = [];
224 dropZone.addEventListener('click', (e) => {
225   if (e.target.closest('label, button')) {
226     e.stopPropagation();
227     return;
228   }
229   fileInput.click();
230 });
231 if (chooseFilesBtn) {
232   chooseFilesBtn.addEventListener('click', () => fileInput.click());
233 }
234 if (takePhotoBtn) {
235   takePhotoBtn.addEventListener('click', () => cameraInput.click());
236 }
237 dropZone.addEventListener('dragover', (e) => {
238   e.preventDefault();
239   dropZone.classList.add('drag-over');
240 });
241 dropZone.addEventListener('dragleave', () => {
242   dropZone.classList.remove('drag-over');
243 });
244 dropZone.addEventListener('drop', (e) => {
245   e.preventDefault();
246   dropZone.classList.remove('drag-over');
247
248   const files = Array.from(e.dataTransfer.files);
249   handleFiles(files);
250 });
251
252 fileInput.addEventListener('change', (e) => {
253   const files = Array.from(e.target.files || []);
254   handleFiles(files);
255   e.target.value = '';
256 });
257
258 cameraInput.addEventListener('change', (e) => {
259   const files = Array.from(e.target.files || []);
260   handleFiles(files);
261   e.target.value = '';
262 });
263 function handleFiles(files) {
264   const validFiles = files.filter(file => {
265     const validTypes = ['image/png', 'image/jpeg', 'image/heic', 'image/heif'];
266     return validTypes.includes(file.type) || file.type.startsWith('image/');
267   });
268   validFiles.forEach(file => {
269     const exists = selectedFiles.some(existingFile =>
270       existingFile.name === file.name &&
271       existingFile.size === file.size &&
272       existingFile.lastModified === file.lastModified
273     );
274
275     if (!exists) {
276       selectedFiles.push(file);
277     }
278   });
279   syncFormInput();
280   updatePreview();
281   updateUploadBtn();
282   updateSelectionSummary();
283 }

```

Σχήμα 5.5.4.1: Διαχείριση πολλαπλών εικόνων και drag-and-drop κατά την προσθήκη γνήσιων υπογραφών.

Πριν από την αποστολή, οι εικόνες μπορούν επίσης να εμφανιστούν τοπικά στον browser. Η προεπισκόπηση δημιουργείται μέσω του FileReader, χωρίς να απαιτείται πρώτα upload στον server. Ο χρήστης έχει έτσι τη δυνατότητα να ελέγξει τις εικόνες που επέλεξε και να αφαιρέσει κάποια πριν ολοκληρώσει την υποβολή.

```

285 function updatePreview() {
286     previewContainer.innerHTML = '';
287
288     if (selectedFiles.length === 0) {
289         return;
290     }
291
292     const grid = document.createElement('div');
293     grid.className = 'row g-3 mb-4';
294
295     selectedFiles.forEach((file, index) => {
296         const reader = new FileReader();
297         reader.onload = (e) => {
298             const col = document.createElement('div');
299             col.className = 'col-6 col-md-4';
300             col.innerHTML = `
301                 <div class="file-preview" data-index="${index}">
302                     
303                     <div class="file-preview-info" title="${file.name}">
304                         ${file.name}
305                     </div>
306                     <div class="file-preview-overlay">
307                         <span class="btn-remove" onclick="removeFile(${index})">
308                             &times;
309                         </span>
310                     </div>
311                 </div>
312             `;
313             grid.appendChild(col);
314         };
315         reader.readAsDataURL(file);
316     });
317
318     previewContainer.appendChild(grid);
319 }

```

Σχήμα 5.5.4.2: Δημιουργία τοπικής προεπισκόπησης των επιλεγμένων εικόνων με χρήση του FileReader.

5.5.5 Προβολή και φιλτράρισμα υπογραφών

Η κεντρική σελίδα προβολής των υπογραφών υλοποιείται στο `app/Views/Signatures/index.php`. Σκοπός της είναι να παρουσιάζει οργανωμένα τις αποθηκευμένες υπογραφές και ταυτόχρονα να δίνει στον χρήστη τη δυνατότητα να περιορίζει τα αποτελέσματα με βάση συγκεκριμένα κριτήρια. Οι υπογραφές εμφανίζονται σε μορφή responsive cards, ώστε η διάταξη να προσαρμόζεται ανάλογα με το διαθέσιμο πλάτος της οθόνης.

Στο πάνω μέρος της σελίδας υπάρχει φόρμα φίλτρων που υποβάλλεται με GET. Ο χρήστης μπορεί να φιλτράρει τις υπογραφές ανά μέλος, κατάσταση, πηγή αποτελέσματος και τελικό αποτέλεσμα, ενώ μπορεί επίσης να επιλέξει πόσες εγγραφές θα εμφανίζονται ανά σελίδα. Οι ήδη επιλεγμένες τιμές διατηρούνται μέσω του πίνακα `$filters`, ώστε το interface να παραμένει συγχρονισμένο με τα αποτελέσματα που εμφανίζονται.

Όπως φαίνεται στο παρακάτω σχήμα, τα φίλτρα δημιουργούνται δυναμικά. Στην περίπτωση των μελών, εκτός από τις επιλογές για όλα τα μέλη και για τον ίδιο τον κάτοχο του λογαριασμού, εμφανίζονται και τα πρόσθετα μέλη που έχουν αποσταλεί στο view μέσω του `$memberOptions`. Παράλληλα, το φίλτρο Result source επιτρέπει τον διαχωρισμό μεταξύ της βασικής ανάλυσης και της εναλλακτικής AI ανάλυσης.

```

69 <div class="glass-panel p-3 p-md-4 mb-4">
70 <form method="get" action="{? site_url('/signatures') ?}" class="row g-2 g-md-3 align-items-end">
71 <div class="col-12 col-md-6 col-lg-3">
72 <label for="member" class="form-label mb-1">Member</label>
73 <select id="member" name="member" class="form-select">
74 <option value="all" <?=$filters['member'] === 'all' ? 'selected' : '' ?>>All members</option>
75 <option value="self" <?=$filters['member'] === 'self' ? 'selected' : '' ?>>Your account only</option>
76 <?php foreach ($memberOptions as $member): ?>
77 <option value="{? esc((string) $member['id']) ?}" <?=$filters['member'] === (string) $member['id'] ? 'selected' : '' ?>>
78 <?=$esc($member['label']) ?>
79 </option>
80 <?php endforeach; ?>
81 </select>
82 </div>
83
84 <div class="col-6 col-lg-2">
85 <label for="status" class="form-label mb-1">Status</label>
86 <select id="status" name="status" class="form-select">
87 <option value="all" <?=$filters['status'] === 'all' ? 'selected' : '' ?>>All</option>
88 <option value="genuine" <?=$filters['status'] === 'genuine' ? 'selected' : '' ?>>Genuine only</option>
89 <option value="tested" <?=$filters['status'] === 'tested' ? 'selected' : '' ?>>Tested only</option>
90 <option value="pending" <?=$filters['status'] === 'pending' ? 'selected' : '' ?>>Pending test</option>
91 </select>
92 </div>
93
94 <div class="col-6 col-lg-2">
95 <label for="result_source" class="form-label mb-1">Result source</label>
96 <select id="result_source" name="result_source" class="form-select" <?=$filters['status'] === 'genuine' ? 'disabled' : '' ?>>
97 <option value="normal" <?=$filters['resultSource'] ?? 'normal' === 'normal' ? 'selected' : '' ?>>Normal detection</option>
98 <option value="ai" <?=$filters['resultSource'] ?? 'normal' === 'ai' ? 'selected' : '' ?>>AI detection</option>
99 </select>
100 </div>
101
102 <div class="col-6 col-lg-2">
103 <label for="result" class="form-label mb-1">Result</label>
104 <select id="result" name="result" class="form-select" <?=$filters['status'] !== 'tested' ? 'disabled' : '' ?>>
105 <option value="all" <?=$filters['result'] === 'all' ? 'selected' : '' ?>>Any</option>
106 <option value="genuine" <?=$filters['result'] === 'genuine' ? 'selected' : '' ?>>Genuine</option>
107 <option value="forged" <?=$filters['result'] === 'forged' ? 'selected' : '' ?>>Forged</option>
108 </select>
109 </div>
110
111 <div class="col-6 col-lg-1">
112 <label for="per_page" class="form-label mb-1">Per page</label>
113 <select id="per_page" name="per_page" class="form-select">
114 <option value="12" <?=(int) $filters['perPage'] === 12 ? 'selected' : '' ?>>12</option>
115 <option value="24" <?=(int) $filters['perPage'] === 24 ? 'selected' : '' ?>>24</option>
116 <option value="48" <?=(int) $filters['perPage'] === 48 ? 'selected' : '' ?>>48</option>
117 </select>
118 </div>
119
120 <div class="col-6 col-lg-1 d-grid">
121 <button type="submit" class="btn btn-primary">Apply</button>

```

Σχήμα 5.5.5.1: Υλοποίηση των φίλτρων για μέλος, κατάσταση και πηγή αποτελέσματος στη λίστα υπογραφών.

Η κατάσταση ορισμένων φίλτρων μεταβάλλεται και δυναμικά μέσω JavaScript. Για παράδειγμα, όταν επιλεγεί η κατάσταση γνήσια, το φίλτρο Result source απενεργοποιείται, καθώς οι γνήσιες reference signatures δεν διαθέτουν αποτέλεσμα ανάλυσης. Αντίστοιχα, το φίλτρο Result ενεργοποιείται μόνο όταν η κατάσταση έχει οριστεί σε tested. Με αυτόν τον τρόπο αποφεύγονται συνδυασμοί φίλτρων που δεν έχουν νόημα για τα δεδομένα που προβάλλονται.

```

173 <script>
174 (function () {
175     const statusSelect = document.getElementById('status');
176     const resultSourceSelect = document.getElementById('result_source');
177     const resultSelect = document.getElementById('result');
178
179     if (!statusSelect || !resultSourceSelect || !resultSelect) {
180         return;
181     }
182
183     const syncResultFilterState = function () {
184         const isTested = statusSelect.value === 'tested';
185         resultSourceSelect.disabled = statusSelect.value === 'genuine';
186         resultSelect.disabled = !isTested;
187         if (!isTested) {
188             resultSelect.value = 'all';
189         }
190     };
191
192     statusSelect.addEventListener('change', syncResultFilterState);
193     syncResultFilterState();
194 })();
195 </script>

```

Σχήμα 5.5.5.2: Δυναμική ενεργοποίηση και απενεργοποίηση φίλτρων με JavaScript ανάλογα με την επιλεγμένη κατάσταση.

Οι υπογραφές εμφανίζονται σε κάρτες τριών στηλών σε μεγάλες οθόνες και προσαρμόζονται αυτόματα σε μικρότερα πλάτη μέσω των κλάσεων col-12, col-md-6 και col-lg-4. Κάθε κάρτα περιλαμβάνει την

εικόνα της υπογραφής, τον τίτλο, το πρόσωπο στο οποίο ανήκει, την κατάστασή της, τα διαθέσιμα αποτελέσματα και βασικές πληροφορίες για το αρχείο.

Τέλος, όταν ο αριθμός των εγγραφών είναι μεγαλύτερος από αυτόν που έχει επιλέξει ο χρήστης για κάθε σελίδα, χρησιμοποιείται pagination μέσω του CodeIgniter pager. Με αυτόν τον τρόπο αποφεύγεται η φόρτωση μεγάλου αριθμού cards σε μία μόνο σελίδα και η πλοήγηση παραμένει πιο εύχρηστη.

Ιδιαίτερη σημασία έχει ο δυναμικός τρόπος με τον οποίο εμφανίζονται τα αποτελέσματα. Αν η υπογραφή έχει χαρακτηριστεί ως γνήσια reference signature, εμφανίζεται ως Genuine. Διαφορετικά, χρησιμοποιείται η κατάσταση της βασικής ανάλυσης. Αν υπάρχουν αποτελέσματα από την κανονική ή την AI ανάλυση, αυτά εμφανίζονται ξεχωριστά με διαφορετικά badges, ώστε να είναι σαφής η προέλευση κάθε αποτελέσματος.

5.5.6 Responsive σχεδίαση και ασφαλής εμφάνιση δεδομένων

Η εφαρμογή σχεδιάστηκε ώστε να μπορεί να χρησιμοποιείται τόσο από υπολογιστή όσο και από κινητή συσκευή. Για τη βασική διάταξη χρησιμοποιούνται οι responsive κλάσεις του Bootstrap, ενώ σε ορισμένες περιπτώσεις έχουν προστεθεί media queries για λειτουργίες που έχουν νόημα μόνο σε μικρότερες οθόνες.

Χαρακτηριστικό παράδειγμα αποτελεί το κουμπί λήψης φωτογραφίας. Σε μεγαλύτερες οθόνες παραμένει κρυφό, ενώ εμφανίζεται όταν το πλάτος της οθόνης αντιστοιχεί σε κινητή συσκευή. Αντίστοιχα, το navigation προσαρμόζεται σε μικρότερα πλάτη ώστε τα στοιχεία του να μπορούν να αναδιπλωθούν χωρίς να υπερβαίνουν τα όρια της οθόνης.

```
45 <style>
46   .mobile-only-camera {
47     display: none;
48   }
49   @media (max-width: 768px) {
50     .mobile-only-camera {
51       display: inline-flex;
52     }
53   }
54   @media (max-width: 991.98px) {
55     .navbar-auth {
56       margin-left: 0;
57       flex-wrap: wrap;
58       justify-content: flex-end;
59       width: 100%;
60       padding-top: 0.75rem;
61     }
62   }
63 </style>
```

Σχήμα 5.5.6: Media queries για την προσαρμογή του navigation και των λειτουργιών κάμερας σε μικρότερες οθόνες.

Επιπλέον, ιδιαίτερη σημασία δόθηκε στην ασφαλή εμφάνιση των δυναμικών δεδομένων μέσα στα views. Καθώς αρκετές τιμές προέρχονται είτε από στοιχεία που έχει εισαγάγει ο χρήστης είτε από εγγραφές της βάσης δεδομένων, δεν ενσωματώνονται απευθείας στον παραγόμενο HTML κώδικα. Για

τον σκοπό αυτό χρησιμοποιείται συστηματικά η συνάρτηση `esc()` του CodeIgniter, η οποία εφαρμόζεται σε πεδία όπως οι τίτλοι και οι περιγραφές των υπογραφών, τα ονόματα των μελών, τα ονόματα αρχείων και τα αποτελέσματα των αναλύσεων. Με αυτόν τον τρόπο, ειδικοί χαρακτήρες που θα μπορούσαν να ερμηνευθούν από τον browser ως HTML μετατρέπονται σε ασφαλή μορφή πριν εμφανιστούν στη σελίδα, περιορίζοντας τον κίνδυνο εισαγωγής ανεπιθύμητου HTML ή JavaScript περιεχομένου.

Παράλληλα, για τη δημιουργία των συνδέσμων της εφαρμογής χρησιμοποιούνται οι helpers `url_to()` και `site_url()`. Η `url_to()` αξιοποιείται όταν ο σύνδεσμος αντιστοιχεί σε ονομασμένη route ή σε συγκεκριμένη μέθοδο controller, ενώ η `site_url()` χρησιμοποιείται για τη δημιουργία URLs με βάση τη βασική διεύθυνση της εφαρμογής. Η χρήση των συγκεκριμένων helpers αποφεύγει τη χειροκίνητη κατασκευή συνδέσμων μέσα στα views και διατηρεί τη δημιουργία των URLs συμβατή με τη ρύθμιση και τη δρομολόγηση της εφαρμογής.

Αντίστοιχη προσοχή δίνεται και στις φόρμες που αποστέλλουν δεδομένα μέσω της μεθόδου POST. Σε αυτές ενσωματώνεται CSRF token μέσω του `csrf_field()` ή των form helpers του CodeIgniter. Το token αποστέλλεται μαζί με τη φόρμα και ελέγχεται από την εφαρμογή κατά την επεξεργασία του αιτήματος. Με αυτόν τον τρόπο παρέχεται προστασία έναντι αιτημάτων CSRF, καθώς μειώνεται ο κίνδυνος εκτέλεσης μη εξουσιοδοτημένων ενεργειών μέσω πλαστών requests που προέρχονται από διαφορετική ιστοσελίδα.

Κεφάλαιο 6ο: Παρουσίαση της Signalyze

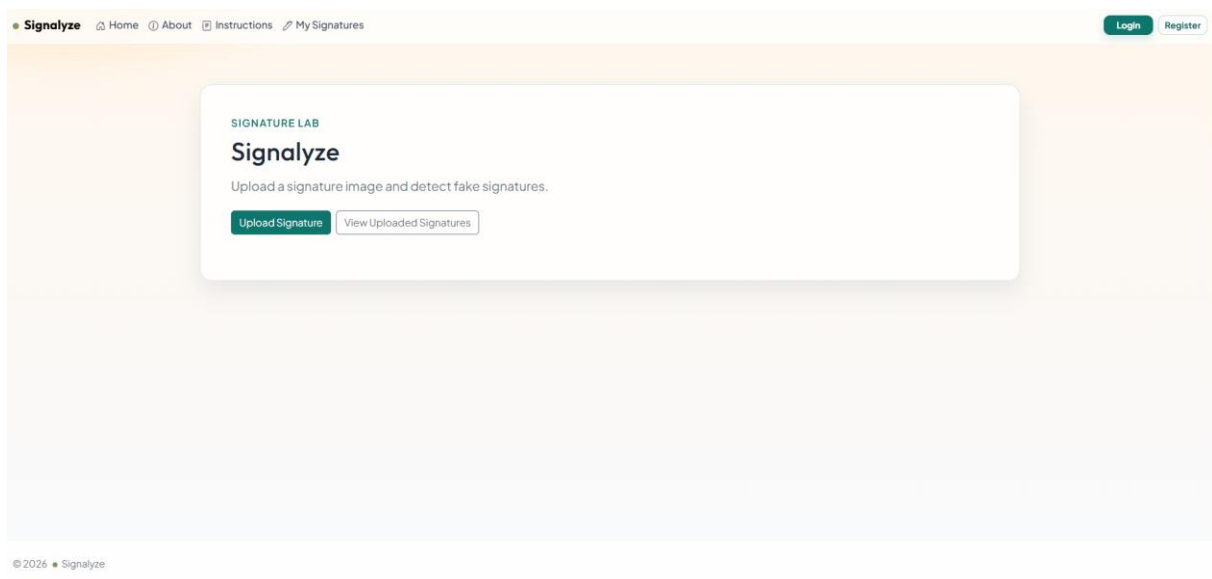
Στο παρόν κεφάλαιο παρουσιάζεται η τελική μορφή της εφαρμογής Signalyze μέσα από τις βασικές οθόνες και τις κύριες λειτουργίες που είναι διαθέσιμες στον χρήστη. Σε αντίθεση με το προηγούμενο κεφάλαιο, όπου δόθηκε έμφαση στη σχεδίαση και την υλοποίηση του συστήματος, εδώ η παρουσίαση επικεντρώνεται στον τρόπο με τον οποίο ο χρήστης αλληλεπιδρά με την εφαρμογή στην πράξη.

Η παρουσίαση ακολουθεί τη βασική ροή χρήσης της Signalyze. Αρχικά παρουσιάζονται οι δημόσιες σελίδες και οι δυνατότητες εγγραφής και σύνδεσης. Στη συνέχεια περιγράφεται η διαδικασία δημιουργίας του συνόλου γνήσιων υπογραφών αναφοράς, η μεταφόρτωση μιας νέας υπογραφής προς εξέταση και η εκτέλεση των διαθέσιμων αναλύσεων. Τέλος, παρουσιάζονται οι λειτουργίες διαχείρισης μελών και οι ρυθμίσεις του προφίλ του χρήστη.

6.1 Αρχική σελίδα

Η αρχική σελίδα αποτελεί το βασικό σημείο εισόδου στην εφαρμογή. Ο σχεδιασμός της είναι λιτός και επικεντρώνεται στις δύο κύριες ενέργειες που μπορεί να πραγματοποιήσει ο χρήστης: τη μεταφόρτωση μιας νέας υπογραφής και την προβολή των ήδη καταχωρισμένων υπογραφών.

Στο επάνω μέρος της σελίδας βρίσκεται η κοινή γραμμή πλοήγησης της Signalyze, μέσω της οποίας παρέχεται πρόσβαση στις σελίδες Home, About, Instructions και My Signatures. Όταν ο χρήστης δεν έχει συνδεθεί, εμφανίζονται επίσης οι επιλογές Login και Register.

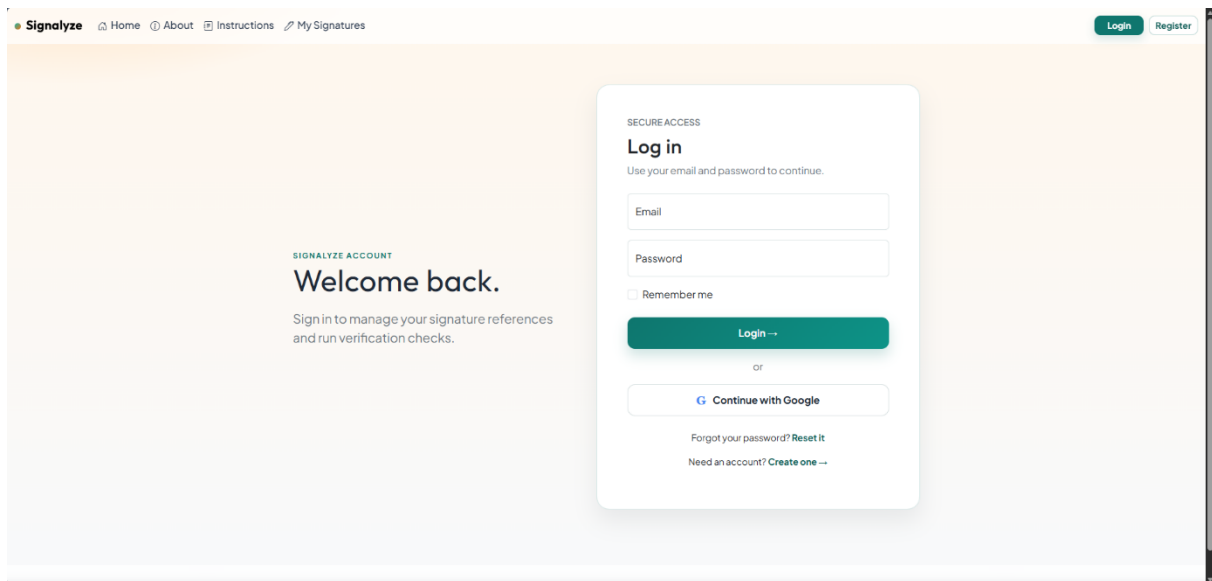


Σχήμα 6.1: Αρχική σελίδα της εφαρμογής Signalyze.

6.2 Σύνδεση χρήστη

Η σελίδα σύνδεσης επιτρέπει την πρόσβαση σε υπάρχον λογαριασμό μέσω email και κωδικού πρόσβασης. Παρέχεται επίσης η επιλογή Remember me, καθώς και σύνδεση μέσω Google.

Στην ίδια οθόνη υπάρχει σύνδεσμος επαναφοράς κωδικού πρόσβασης και δυνατότητα μετάβασης στη σελίδα δημιουργίας νέου λογαριασμού. Μετά την επιτυχημένη σύνδεση, ο χρήστης αποκτά πρόσβαση στις προσωπικές υπογραφές, στα μέλη που έχει καταχωρίσει και στις ρυθμίσεις του προφίλ του.

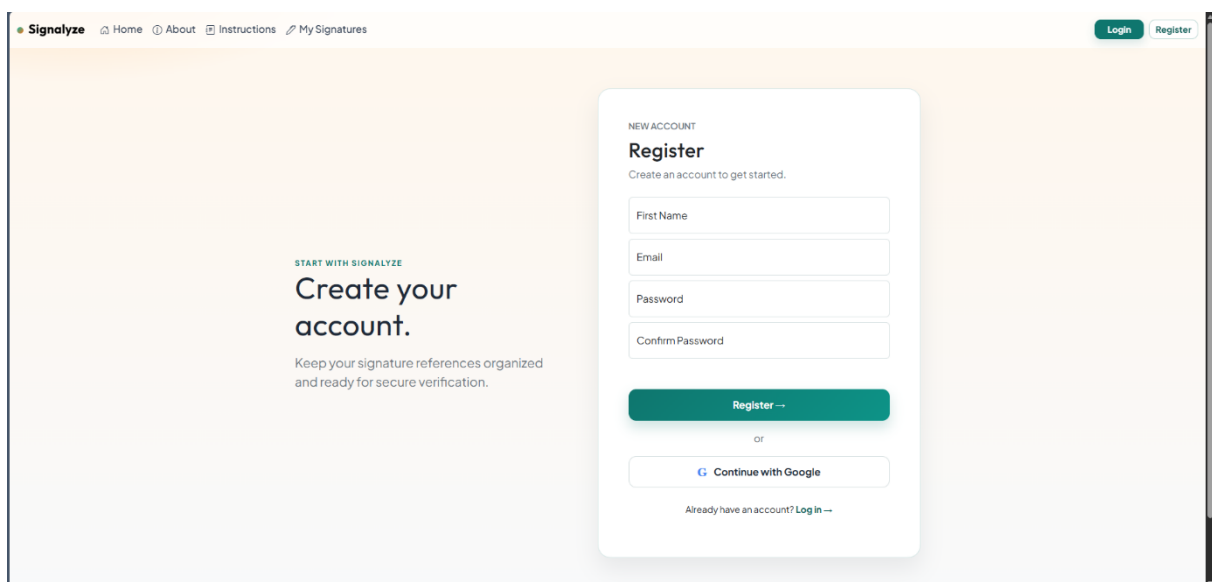


Σχήμα 6.2: Σελίδα σύνδεσης στη Signalyze.

6.3 Δημιουργία νέου λογαριασμού

Η εγγραφή νέου χρήστη πραγματοποιείται μέσω της σελίδας Register. Ο χρήστης συμπληρώνει το όνομά του, τη διεύθυνση email και τον επιθυμητό κωδικό πρόσβασης, τον οποίο καλείται να επιβεβαιώσει σε δεύτερο πεδίο.

Εναλλακτικά, μπορεί να δημιουργήσει λογαριασμό ή να συνδεθεί χρησιμοποιώντας το Google Sign-In. Η δυνατότητα αυτή περιορίζει τα βήματα που απαιτούνται για τη δημιουργία λογαριασμού και παρέχει έναν δεύτερο τρόπο αυθεντικοποίησης.



Σχήμα 6.3: Φόρμα δημιουργίας νέου λογαριασμού.

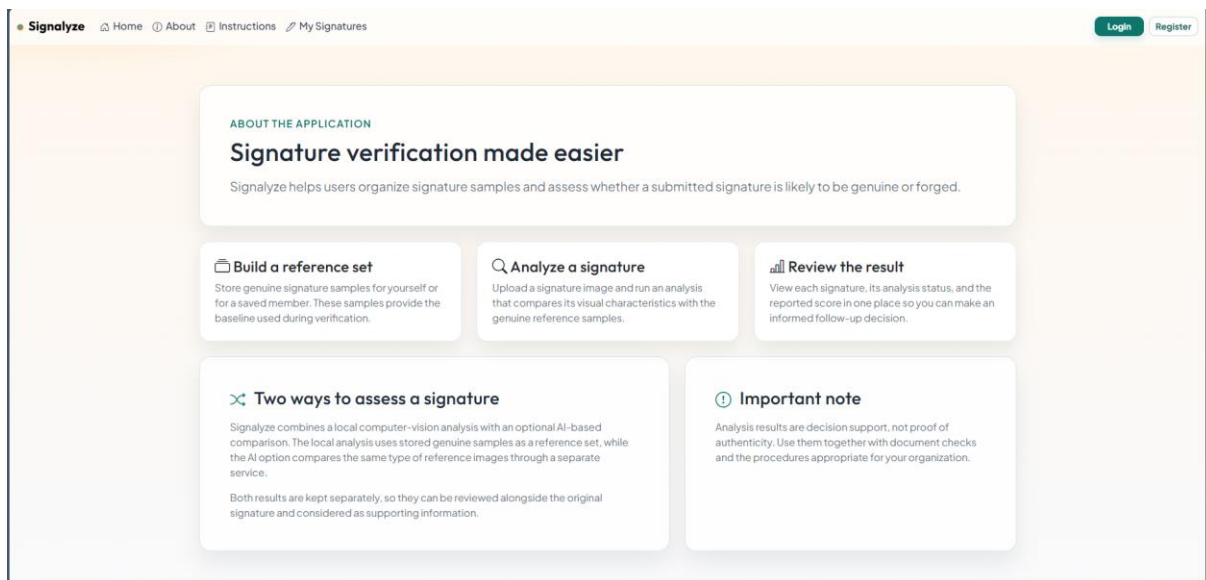
6.4 Σελίδα πληροφοριών της εφαρμογής

Η σελίδα About παρουσιάζει συνοπτικά τον σκοπό της Signalyze και τις βασικές δυνατότητές της. Ο χρήστης ενημερώνεται ότι η εφαρμογή χρησιμοποιείται για την οργάνωση γνήσιων δειγμάτων

υπογραφών και για την αξιολόγηση νέων υπογραφών ως προς το αν παρουσιάζουν χαρακτηριστικά γνήσιας ή πλαστής υπογραφής.

Παρουσιάζονται επίσης οι τρεις βασικές λειτουργίες της εφαρμογής: η δημιουργία ενός συνόλου αναφοράς, η ανάλυση μιας νέας υπογραφής και η εξέταση του αποτελέσματος. Στην ίδια σελίδα διευκρινίζεται ότι η Signalyze διαθέτει δύο διαφορετικούς τρόπους αξιολόγησης: την κύρια ανάλυση που βασίζεται στο τοπικό σύστημα και την προαιρετική αξιολόγηση μέσω OpenAI.

Ιδιαίτερη σημασία έχει η σημείωση ότι τα αποτελέσματα της εφαρμογής λειτουργούν υποστηρικτικά και δεν αποτελούν από μόνα τους απόδειξη αυθεντικότητας μιας υπογραφής.



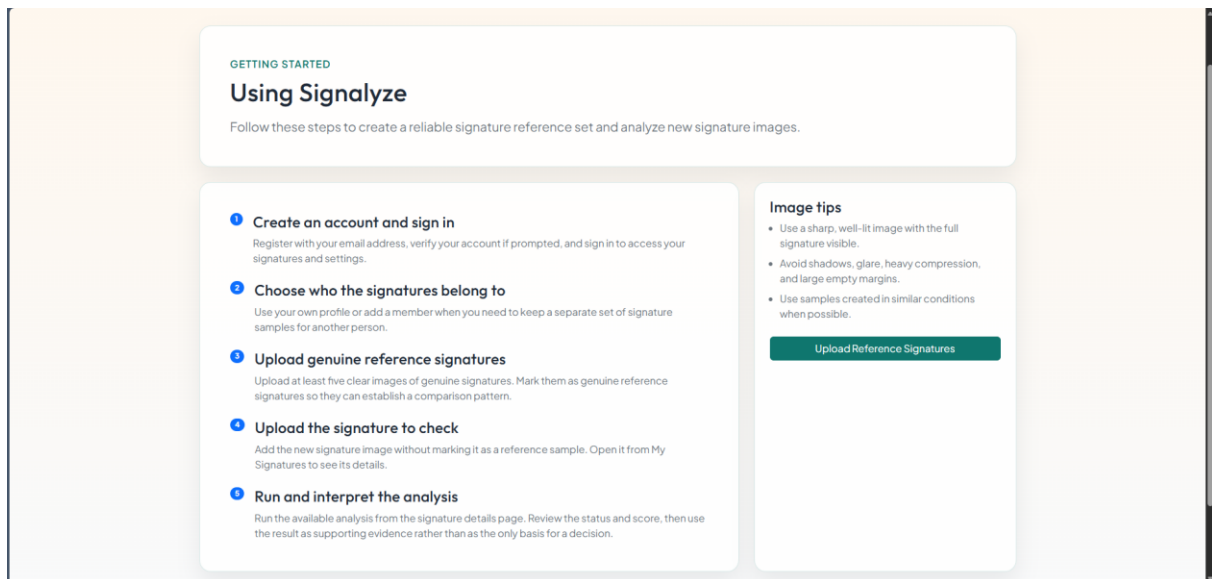
Σχήμα 6.4: Σελίδα πληροφοριών και συνοπτικής παρουσίασης των δυνατοτήτων της Signalyze.

6.5 Οδηγίες χρήσης

Η σελίδα Instructions παρέχει στον χρήστη μία συνοπτική καθοδήγηση για τη σωστή χρήση της εφαρμογής. Η διαδικασία παρουσιάζεται σε διαδοχικά βήματα, ξεκινώντας από τη δημιουργία λογαριασμού και την επιλογή του προσώπου στο οποίο ανήκουν οι υπογραφές.

Στη συνέχεια ο χρήστης καθοδηγείται στη μεταφόρτωση τουλάχιστον πέντε γνήσιων υπογραφών αναφοράς και έπειτα στην εισαγωγή της υπογραφής που επιθυμεί να εξετάσει. Το τελευταίο βήμα αφορά την εκτέλεση της ανάλυσης και την ερμηνεία του αποτελέσματος.

Στη δεξιά πλευρά της σελίδας παρέχονται επίσης βασικές οδηγίες σχετικά με την ποιότητα των εικόνων. Προτείνεται η χρήση καθαρών και καλά φωτισμένων εικόνων, η αποφυγή έντονων σκιών ή αντανακλάσεων και, όπου είναι δυνατό, η χρήση δειγμάτων που έχουν ληφθεί υπό παρόμοιες συνθήκες.



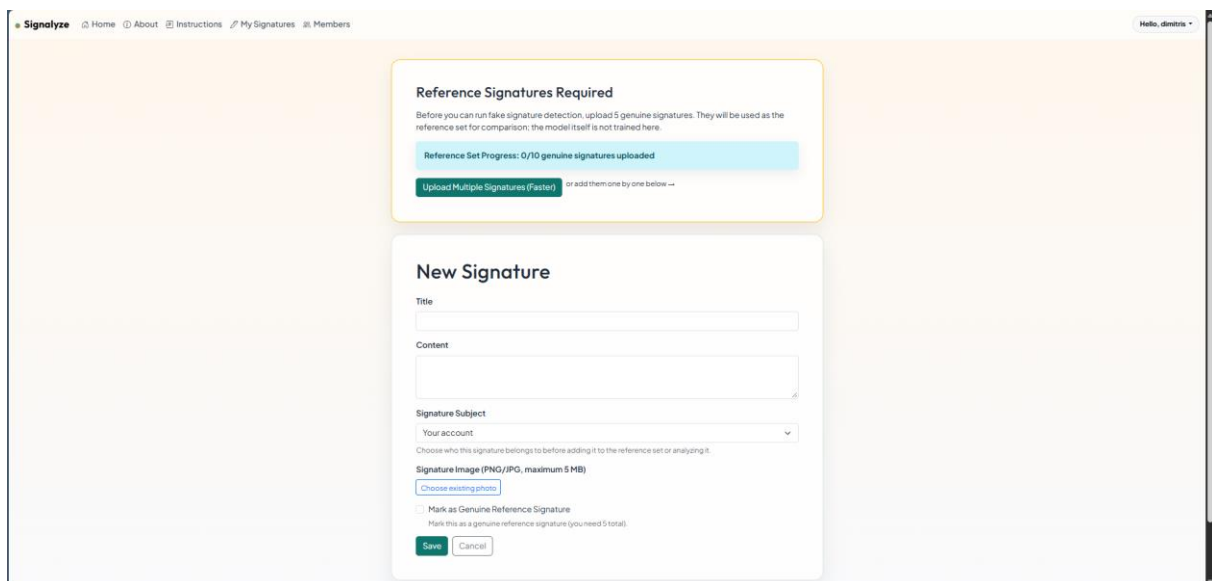
Σχήμα 6.5: Σελίδα οδηγιών χρήσης και προτάσεων για την ποιότητα των εικόνων.

6.6 Προσθήκη νέας υπογραφής

Η σελίδα New Signature χρησιμοποιείται για την εισαγωγή μίας νέας υπογραφής στην εφαρμογή. Ο χρήστης μπορεί να ορίσει τίτλο και περιγραφή, να επιλέξει το πρόσωπο στο οποίο ανήκει η υπογραφή και να μεταφορτώσει την αντίστοιχη εικόνα.

Όταν ο λογαριασμός δεν διαθέτει ακόμη το ελάχιστο πλήθος γνήσιων υπογραφών αναφοράς, εμφανίζεται σχετική ενημέρωση στο επάνω μέρος της σελίδας. Ο χρήστης πληροφορείται για την πρόοδο του συνόλου αναφοράς και μπορεί είτε να προσθέσει τις γνήσιες υπογραφές μία προς μία είτε να μεταβεί στη λειτουργία μαζικής μεταφόρτωσης.

Η επιλογή Mark as γνήσια υπογραφή αναφοράς χρησιμοποιείται όταν η εικόνα που εισάγεται αποτελεί γνήσιο δείγμα και πρέπει να συμμετέχει στο σύνολο αναφοράς του αντίστοιχου προσώπου.



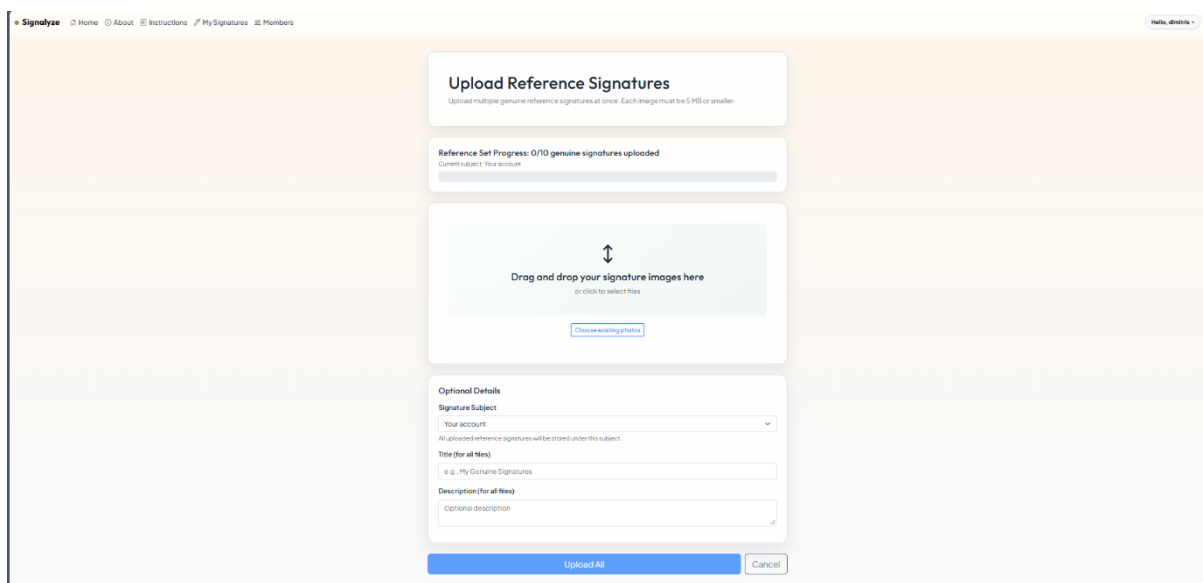
Σχήμα 6.6: Σελίδα εισαγωγής νέας υπογραφής και ενημέρωση για την πρόοδο του συνόλου αναφοράς.

6.7 Μαζική μεταφόρτωση γνήσιων υπογραφών

Για τη γρηγορότερη δημιουργία του συνόλου αναφοράς παρέχεται ξεχωριστή οθόνη μαζικής μεταφόρτωσης. Ο χρήστης μπορεί να επιλέξει ταυτόχρονα περισσότερες από μία εικόνες ή να τις μεταφέρει απευθείας στην περιοχή drag-and-drop.

Στο επάνω μέρος εμφανίζεται η τρέχουσα πρόοδος του συνόλου αναφοράς, με ανώτατο όριο τις δέκα γνήσιες υπογραφές για κάθε πρόσωπο. Στο κάτω μέρος μπορούν να οριστούν προαιρετικά κοινός τίτλος και περιγραφή για τα αρχεία που πρόκειται να προστεθούν, ενώ μέσω του πεδίου Signature Subject επιλέγεται το πρόσωπο στο οποίο θα συσχετιστούν.

Η δυνατότητα αυτή είναι ιδιαίτερα χρήσιμη κατά την αρχική δημιουργία ενός συνόλου αναφοράς, καθώς μειώνει τον αριθμό των ξεχωριστών ενεργειών που πρέπει να πραγματοποιήσει ο χρήστης.

The screenshot shows the 'Upload Reference Signatures' page in the Signalyze application. At the top, there's a navigation bar with links to Home, About, Instructions, My Signatures, and Members. The main content area has a light orange background. It features a section titled 'Upload Reference Signatures' with a subtext: 'Upload multiple genuine reference signatures at once. Each image must be 5 MB or smaller.' Below this is a progress bar labeled 'Reference Set Progress: 0/10 genuine signatures uploaded' and 'Current subject: Your account'. A large light blue box in the center contains a double-headed arrow icon and the text 'Drag and drop your signature images here' with a link 'or click to select files' and a 'Choose existing photos' button. At the bottom, there's an 'Optional Details' section with a 'Signature Subject' dropdown (set to 'Your account'), a 'Title (for all files)' text field (with example 'e.g., My Genuine Signatures'), and a 'Description (for all files)' text area (with placeholder 'Optional description'). At the very bottom are 'Upload All' and 'Cancel' buttons.

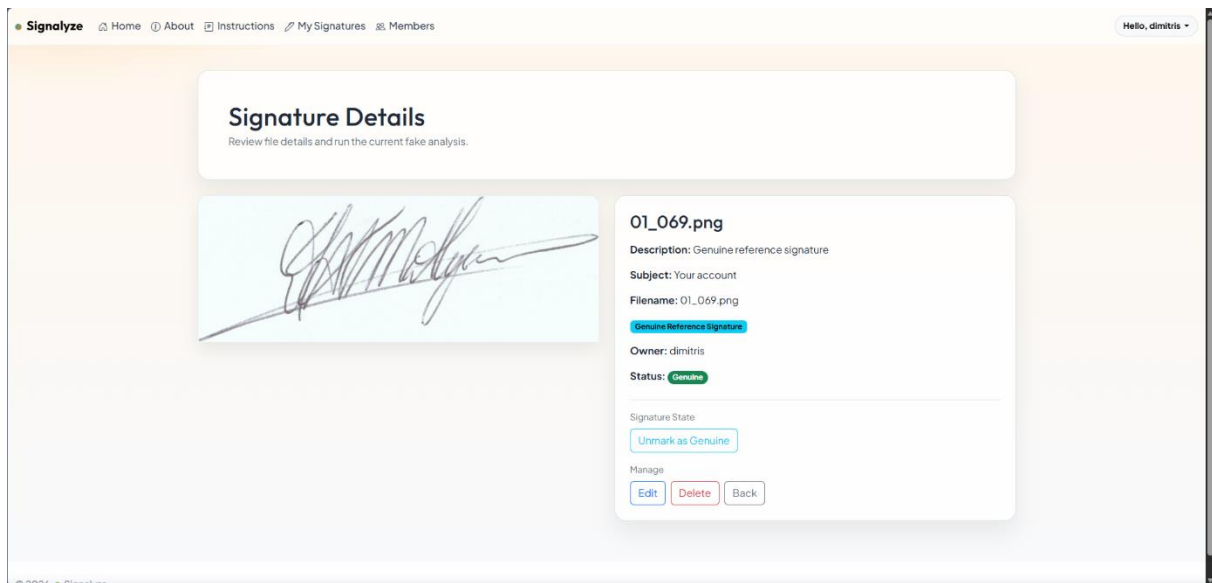
Σχήμα 6.7: Σελίδα μαζικής μεταφόρτωσης γνήσιων υπογραφών αναφοράς.

6.8 Προβολή γνήσιας υπογραφής αναφοράς

Μετά τη μεταφόρτωση μιας υπογραφής, ο χρήστης μπορεί να ανοίξει τη σελίδα λεπτομερειών της. Στην περίπτωση μιας γνήσιας υπογραφής αναφοράς εμφανίζονται η εικόνα, ο τίτλος, η περιγραφή, το πρόσωπο στο οποίο ανήκει και το όνομα του αρχείου.

Η ένδειξη γνήσια υπογραφή αναφοράς και το αντίστοιχο status κάνουν σαφή τον ρόλο της συγκεκριμένης εικόνας στο σύνολο αναφοράς. Από την ίδια σελίδα ο χρήστης μπορεί να αφαιρέσει τον χαρακτηρισμό της ως γνήσιας, καθώς και να επεξεργαστεί ή να διαγράψει την εγγραφή.

Οι γνήσιες υπογραφές αναφοράς δεν χρησιμοποιούνται ως εξεταζόμενες υπογραφές και επομένως δεν εμφανίζονται σε αυτές οι ενέργειες εκτέλεσης ανάλυσης.



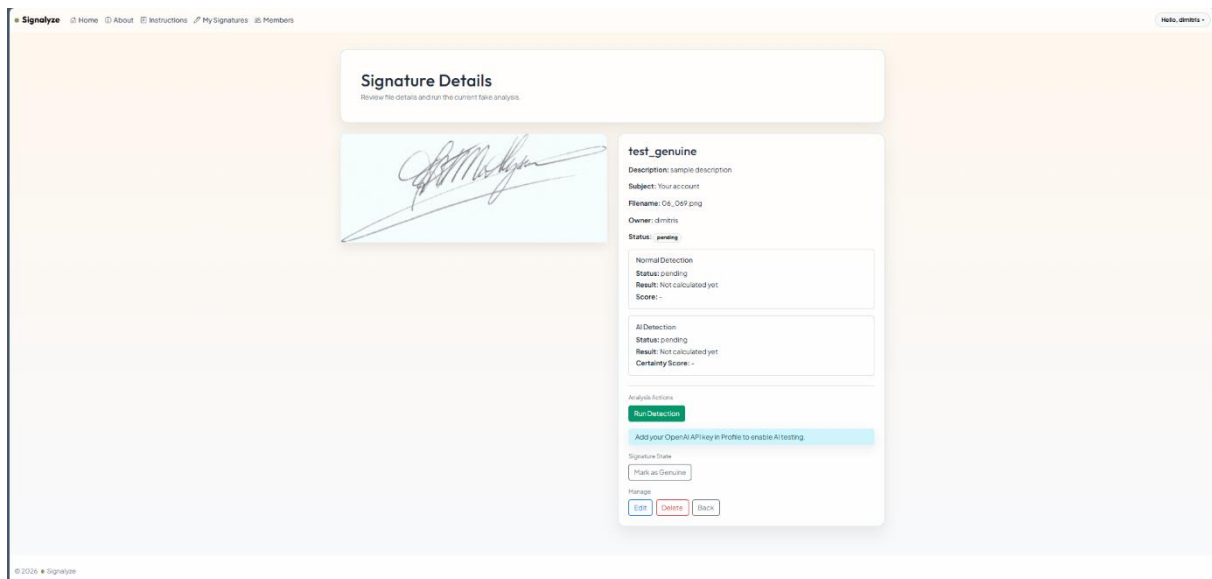
Σχήμα 6.8: Σελίδα λεπτομερειών γνήσιας υπογραφής αναφοράς.

6.9 Υπογραφή πριν από την εκτέλεση της ανάλυσης

Όταν μια υπογραφή έχει προστεθεί ως εξεταζόμενη υπογραφή, στη σελίδα λεπτομερειών εμφανίζεται αρχικά με κατάσταση pending. Πριν από την εκτέλεση της ανάλυσης, τα πεδία των αποτελεσμάτων δεν περιέχουν ακόμη τιμές.

Στην ενότητα Analysis Actions εμφανίζεται το κουμπί Run Detection, μέσω του οποίου ξεκινά η βασική διαδικασία ανάλυσης. Εφόσον ο χρήστης έχει αποθηκεύσει OpenAI API key στο προφίλ του, εμφανίζεται και η αντίστοιχη επιλογή για AI testing. Αν δεν υπάρχει διαθέσιμο API key, η εφαρμογή ενημερώνει τον χρήστη ότι πρέπει πρώτα να το καταχωρίσει στις ρυθμίσεις του προφίλ.

Στην ίδια οθόνη παρέχονται επίσης οι δυνατότητες χαρακτηρισμού της υπογραφής ως γνήσιας, επεξεργασίας ή διαγραφής της.



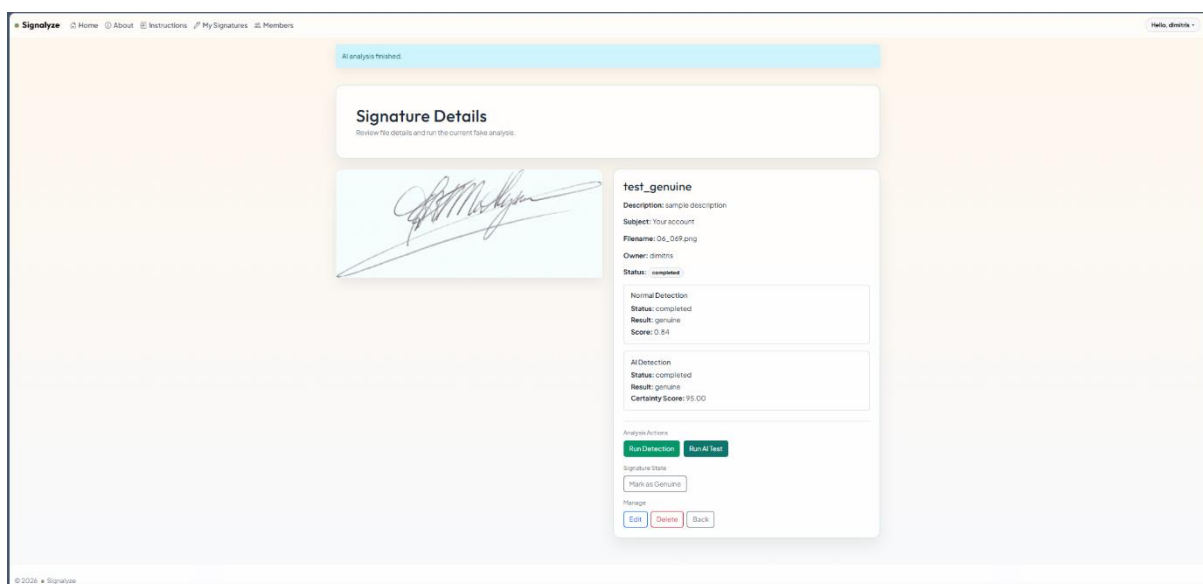
Σχήμα 6.9: Σελίδα εξεταζόμενης υπογραφής πριν από την εκτέλεση των αναλύσεων.

6.10 Προβολή αποτελεσμάτων ανάλυσης

Μετά την ολοκλήρωση της ανάλυσης, η κατάσταση της υπογραφής μεταβάλλεται σε completed και τα αποτελέσματα εμφανίζονται στην ίδια σελίδα. Η κύρια μέθοδος παρουσιάζεται στην περιοχή Normal Detection, όπου εμφανίζονται η κατάσταση της διαδικασίας, το τελικό αποτέλεσμα και ο βαθμός που προέκυψε.

Σε ξεχωριστή περιοχή παρουσιάζεται το αποτέλεσμα του AI Detection, όταν έχει πραγματοποιηθεί η εναλλακτική αξιολόγηση μέσω OpenAI. Στην περίπτωση αυτή εμφανίζονται το αποτέλεσμα και ο βαθμός βεβαιότητας που επιστρέφεται από την υπηρεσία.

Η ξεχωριστή παρουσίαση των δύο αποτελεσμάτων επιτρέπει στον χρήστη να γνωρίζει με ποια μέθοδο παρήχθη κάθε ένδειξη. Στο συγκεκριμένο παράδειγμα, τόσο η κύρια ανάλυση όσο και η AI αξιολόγηση χαρακτήρισαν την υπογραφή ως γνήσια.



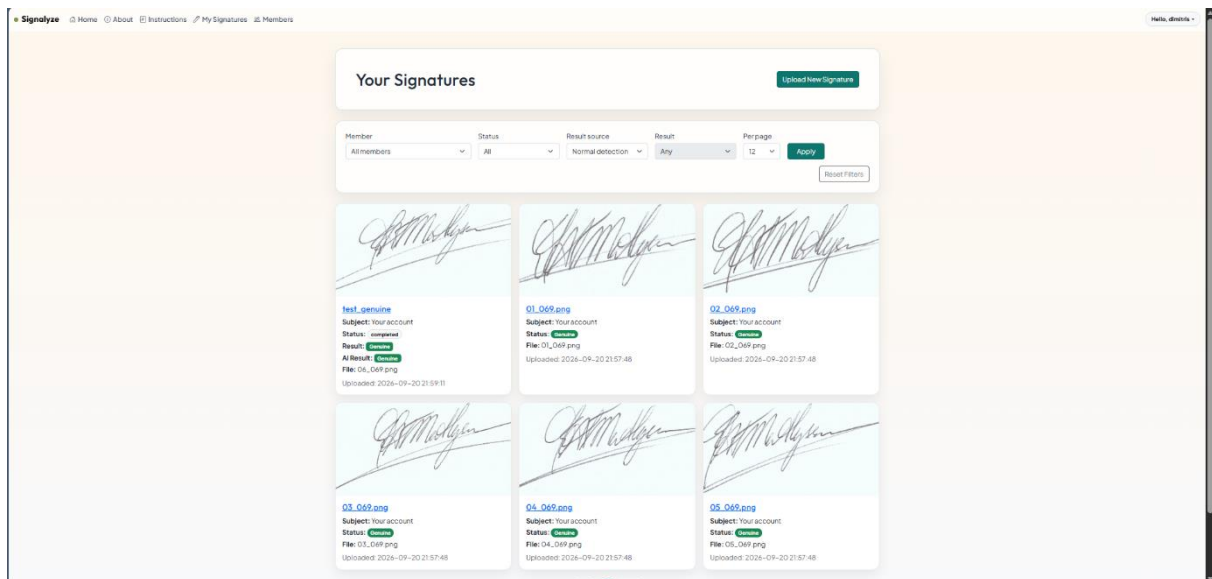
Σχήμα 6.10: Σελίδα λεπτομερειών μετά την ολοκλήρωση της βασικής και της AI ανάλυσης.

6.11 Λίστα υπογραφών

Η σελίδα My Signatures συγκεντρώνει όλες τις υπογραφές που έχουν καταχωριστεί στον λογαριασμό. Κάθε υπογραφή εμφανίζεται σε ξεχωριστή κάρτα με την εικόνα της, τον τίτλο, το πρόσωπο στο οποίο ανήκει, την κατάσταση και, όπου υπάρχουν, τα αποτελέσματα των αναλύσεων.

Στο επάνω μέρος της σελίδας παρέχονται φίλτρα που επιτρέπουν την επιλογή συγκεκριμένου μέλους, κατάστασης, πηγής αποτελέσματος και τελικού χαρακτηρισμού. Παρέχεται επίσης επιλογή του αριθμού εγγραφών ανά σελίδα και δυνατότητα επαναφοράς όλων των φίλτρων.

Με αυτόν τον τρόπο ο χρήστης μπορεί να διαχειρίζεται ευκολότερα μεγαλύτερο αριθμό υπογραφών και να εντοπίζει γρήγορα, για παράδειγμα, μόνο τις γνήσιες υπογραφές αναφοράς ή μόνο τις υπογραφές που έχουν ήδη αναλυθεί.



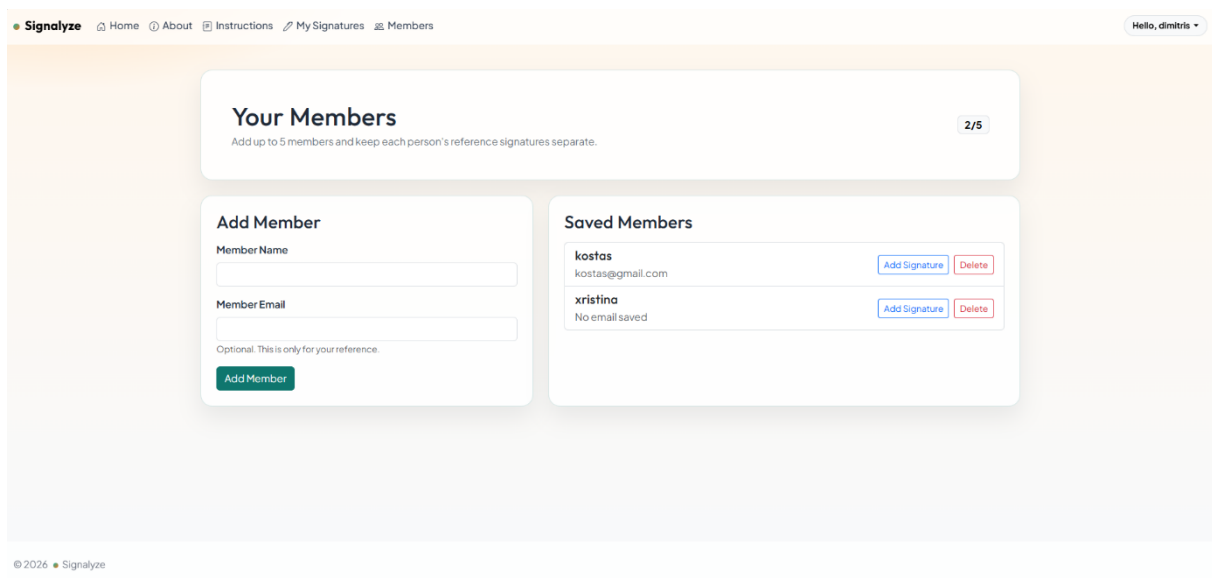
Σχήμα 6.11: Σελίδα συγκεντρωτικής προβολής και φιλτραρίσματος των υπογραφών.

6.12 Διαχείριση μελών

Η Signalyze επιτρέπει τη δημιουργία πρόσθετων μελών μέσα στον ίδιο λογαριασμό, ώστε να διατηρούνται ξεχωριστά σύνολα αναφοράς για διαφορετικά πρόσωπα. Στη σελίδα Members ο χρήστης μπορεί να καταχωρίσει το όνομα ενός νέου μέλους και, προαιρετικά, τη διεύθυνση email του.

Στη δεξιά πλευρά εμφανίζονται τα ήδη καταχωρισμένα μέλη. Για κάθε μέλος παρέχεται άμεση δυνατότητα προσθήκης νέας υπογραφής ή διαγραφής της εγγραφής. Στο επάνω μέρος εμφανίζεται επίσης ο αριθμός των χρησιμοποιούμενων θέσεων, καθώς κάθε λογαριασμός μπορεί να διαθέτει έως πέντε πρόσθετα μέλη.

Η λειτουργία αυτή επιτρέπει σε έναν χρήστη να διαχειρίζεται ανεξάρτητες συλλογές γνήσιων υπογραφών χωρίς να χρειάζεται να δημιουργούνται ξεχωριστοί λογαριασμοί.



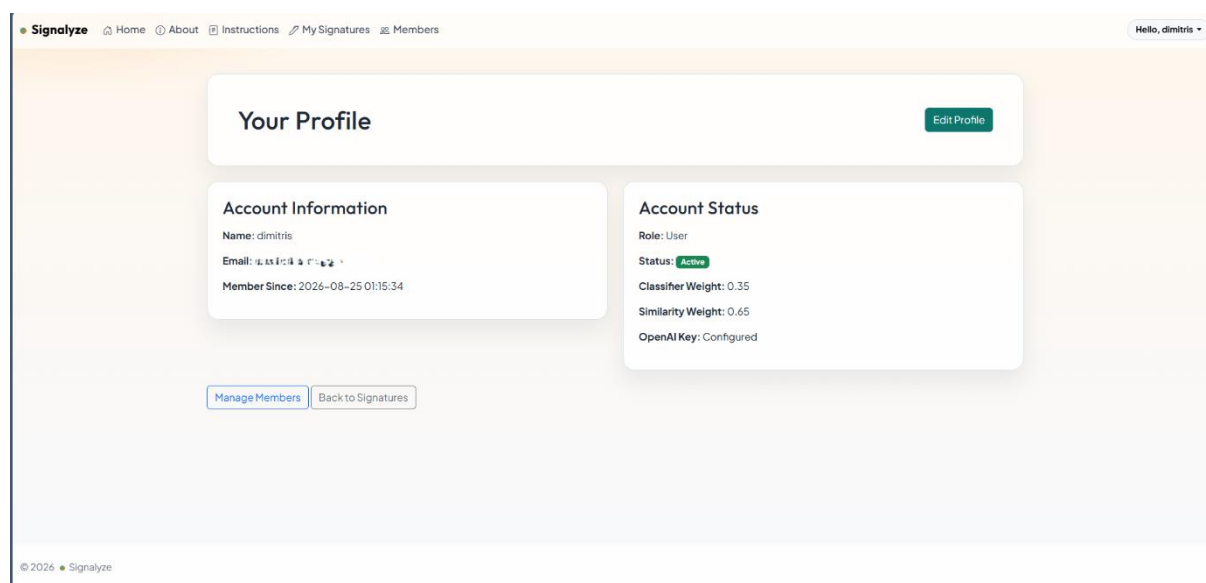
Σχήμα 6.12: Σελίδα δημιουργίας και διαχείρισης πρόσθετων μελών.

6.13 Προβολή προφίλ

Η σελίδα Profile συγκεντρώνει τις βασικές πληροφορίες του λογαριασμού και τις τρέχουσες ρυθμίσεις της ανάλυσης. Στην ενότητα Account Information εμφανίζονται το όνομα, η διεύθυνση email και η ημερομηνία δημιουργίας του λογαριασμού.

Στην ενότητα Account Status παρουσιάζονται η κατάσταση του λογαριασμού, τα δύο βάρη που χρησιμοποιούνται στην κύρια ανάλυση και η κατάσταση του OpenAI API key. Στο συγκεκριμένο παράδειγμα τα βάρη έχουν οριστεί σε 0,35 για την πιθανότητα του ταξινομητή και 0,65 για την συνημιτονική ομοιότητα, ενώ εμφανίζεται επίσης ένδειξη ότι έχει καταχωριστεί OpenAI API key.

Από την ίδια σελίδα ο χρήστης μπορεί να μεταβεί στην επεξεργασία του προφίλ ή στη διαχείριση των πρόσθετων μελών του.



Σχήμα 6.13: Σελίδα προβολής των πληροφοριών και των ρυθμίσεων του προφίλ.

6.14 Επεξεργασία προφίλ και παραμέτρων ανάλυσης

Μέσω της επιλογής Edit Profile ο χρήστης μπορεί να μεταβάλλει τα στοιχεία και τις ρυθμίσεις του λογαριασμού του. Εκτός από το όνομα, παρέχεται η δυνατότητα αλλαγής των βαρών που συμμετέχουν στον υπολογισμό του τελικού βαθμού της βασικής ανάλυσης.

Οι δύο τιμές αντιστοιχούν στην πιθανότητα που προκύπτει από τον ταξινομητή και στην συνημιτονική ομοιότητα αντίστοιχα, ενώ το άθροισμά τους πρέπει να είναι ίσο με 1. Οι προεπιλεγμένες τιμές της εφαρμογής είναι 0,35 και 0,65, σύμφωνα με τα αποτελέσματα της πειραματικής μελέτης.

Στην ίδια σελίδα υπάρχει η ενότητα AI Testing, μέσω της οποίας ο χρήστης μπορεί να καταχωρίσει το προσωπικό του OpenAI API key ώστε να ενεργοποιηθεί η εναλλακτική αξιολόγηση. Αν υπάρχει ήδη αποθηκευμένο key, εμφανίζεται μόνο σχετική ένδειξη και όχι η πλήρης τιμή του. Παρέχεται επίσης δυνατότητα διαγραφής του αποθηκευμένου API key.

Σχήμα 6.14: Σελίδα επεξεργασίας προφίλ, βαρών ανάλυσης και OpenAI API key.

Με τις παραπάνω οθόνες παρουσιάζεται η βασική ροή λειτουργίας της Signalize από την πλευρά του χρήστη, από τη δημιουργία λογαριασμού και την καταχώριση των γνήσιων δειγμάτων έως την εκτέλεση της ανάλυσης και την προβολή των αποτελεσμάτων. Παράλληλα, οι λειτουργίες διαχείρισης μελών και παραμετροποίησης του προφίλ επιτρέπουν την προσαρμογή της εφαρμογής σε διαφορετικά πρόσωπα και διαφορετικές ανάγκες χρήσης.

Κεφάλαιο 7ο: Συμπεράσματα και Μελλοντικές Επεκτάσεις

Στο κεφάλαιο αυτό παρουσιάζονται τα συμπεράσματα της εργασίας και εξετάζονται οι κύριες δυνατότητες μελλοντικής ανάπτυξης της Signalize. Η εργασία περιλάμβανε τόσο το πειραματικό στάδιο ανίχνευσης πιθανών πλαστών υπογραφών όσο και την ανάπτυξη μιας ολοκληρωμένης διαδικτυακής εφαρμογής, μέσω της οποίας ο χρήστης μπορεί να οργανώνει το σύνολο αναφοράς των γνήσιων υπογραφών του και να αναλύει νέα δείγματα.

7.1 Συμπεράσματα

Ο κύριος στόχος αυτής της διατριβής ήταν η δημιουργία μιας διαδικτυακής εφαρμογής για την ανίχνευση πιθανών πλαστών χειρόγραφων υπογραφών χρησιμοποιώντας την ανάλυση εικόνας και τις τεχνικές μηχανικής μάθησης. Η προκύπτουσα εφαρμογή ονομάζεται Signalize και περιλαμβάνει όλα τα στάδια της διαδικασίας, από την καταγραφή δειγμάτων μιας γνήσιας υπογραφής μέχρι την ανίχνευση ενός νέου δείγματος είτε ως γνήσιο είτε ως πλαστό.

Στο πλαίσιο της πειραματικής μελέτης δοκιμάστηκαν διάφοροι συνδυασμοί συνελκτικών νευρωνικών δικτύων για την εξαγωγή χαρακτηριστικών και αλγορίθμων μηχανικής μάθησης για την ταξινόμηση των υπογραφών. Από αυτούς επιλέχθηκε ο συνδυασμός DenseNet169 και Naïve Bayes, καθώς παρουσίασε τα καλύτερα αποτελέσματα στο συγκεκριμένο πειραματικό πλαίσιο. Στη συνέχεια αξιολογήθηκε ο συνδυασμός του αποτελέσματος του ταξινομητή με τη μέση συνημιτονική ομοιότητα μεταξύ της εξεταζόμενης υπογραφής και των δειγμάτων αναφοράς.

Διαπιστώθηκε ότι το καλύτερο αποτέλεσμα επιτυγχάνεται όταν αποδίδεται βάρος 35% στην πιθανότητα του ταξινομητή και 65% στην συνημιτονική ομοιότητα. Με αυτόν τον συνδυασμό επιτεύχθηκε ακρίβεια

62,38% στο σύνολο αξιολόγησης. Το αποτέλεσμα αυτό υποδεικνύει ότι, στο συγκεκριμένο πειραματικό πλαίσιο, η άμεση σύγκριση της εξεταζόμενης υπογραφής με τα γνήσια δείγματα του ίδιου ατόμου παρέχει χρήσιμες πρόσθετες πληροφορίες.

Εκτός από το πειραματικό μέρος της εργασίας μια σημαντική πτυχή ήταν η ενσωμάτωση του αλγορίθμου ανάλυσης υπογραφών σε μια ολοκληρωμένη διαδικτυακή εφαρμογή. Η εφαρμογή Signalyze επιτρέπει τη δημιουργία λογαριασμού, την οργάνωση του συνόλου αναφοράς των γνήσιων υπογραφών, την προσθήκη διαφόρων ατόμων στη βάση δεδομένων μέσω της δυνατότητας "Μέλη" και την αποστολή νέων υπογραφών είτε από αρχείο είτε χρησιμοποιώντας την κάμερα της κινητής συσκευής. Για κάθε άτομο αποθηκεύεται ένα ξεχωριστό σύνολο αναφοράς το οποίο επιτρέπει τη διαχείριση περισσότερων από ενός συνόλων υπογραφών χρησιμοποιώντας τον ίδιο λογαριασμό.

Επίσης η Signalyze επιτρέπει την προσαρμογή των βαρών των συστατικών του τελικού σκορ. Επιπλέον έχει υλοποιηθεί μια επιλογή για την αξιολόγηση της υπογραφής μέσω του API του OpenAI. Αυτή η λειτουργικότητα λειτουργεί ανεξάρτητα από το κύριο μηχανισμό ανάλυσης υπογραφών και παρουσιάζει το δικό της αποτέλεσμα ώστε να καταστούν ξεκάθαρα οι πηγές των εκτιμήσεων.

Επίσης η διαδικασία ανάπτυξης της εφαρμογής έδειξε ότι η πρακτική υλοποίηση ενός συστήματος επαλήθευσης υπογραφών περιλαμβάνει όχι μόνο την επιλογή ενός κατάλληλου μοντέλου μηχανικής μάθησης αλλά και τη διαχείριση των εικόνων, την οργάνωση των δειγμάτων αναφοράς των γνήσιων υπογραφών, την αλληλεπίδραση μεταξύ του PHP back-end και του συστήματος Python, την ασφαλή διαχείριση των δεδομένων και την παρουσίαση του αποτελέσματος με κατανοητό τρόπο για τον χρήστη.

Αν και τα αποτελέσματα του πειράματος δείχνουν ότι η προτεινόμενη προσέγγιση μπορεί, σε κάποιο βαθμό, να διακρίνει μεταξύ γνήσιων και πλαστών υπογραφών, η επιτευχθείσα ακρίβεια δεν είναι επαρκής ώστε να αποτελεί τον μοναδικό τρόπο επιβεβαίωσης της γνησιότητας μιας υπογραφής. Επομένως, η Signalyze θα πρέπει να θεωρείται συμπληρωματικό εργαλείο αξιολόγησης υπογραφών.

Συνολικά η εργασία πέτυχε τον κύριο στόχο της καθώς συνέδεσε την πειραματική εξερεύνηση διαφόρων αλγορίθμων ανάλυσης υπογραφών με την υλοποίηση μιας ολοκληρωμένης διαδικτυακής εφαρμογής. Η προκύπτουσα εφαρμογή είναι ένα λειτουργικό πρωτότυπο για μελλοντικές βελτιώσεις τόσο στην ανάλυση όσο και στην λειτουργικότητα της εφαρμογής.

7.2 Μελλοντικές Αναβαθμίσεις

Αυτή η εργασία επικεντρώνεται στην ανάπτυξη ενός λειτουργικού πρωτοτύπου για την εκτός σύνδεσης επαλήθευση χειρόγραφων υπογραφών και την ενσωμάτωση του αλγορίθμου ανάλυσης σε μια ολοκληρωμένη διαδικτυακή εφαρμογή. Ωστόσο η μελλοντική ανάπτυξη της Signalyze μπορεί να περιλαμβάνει κατευθύνσεις που απαιτούν πολύ περισσότερες έρευνες και τεχνικές προσπάθειες.

Πρώτα απ' όλα, βασική κατεύθυνση για τη μελλοντική ανάπτυξη είναι η δημιουργία ενός συστήματος πολύ μεγαλύτερης κλίμακας. Για τον σκοπό αυτό θα απαιτηθεί η συλλογή ή η εξασφάλιση πρόσβασης σε μεγαλύτερο σύνολο δεδομένων, το οποίο θα περιλαμβάνει περισσότερους υπογράφοντες, περισσότερα γνήσια δείγματα ανά άτομο και διαφορετικούς τύπους πλαστογράφησης. Ένα τέτοιο σύνολο θα επιτρέψει πιο αξιόπιστη αξιολόγηση της ικανότητας γενίκευσης του μοντέλου σε πρόσωπα και συνθήκες που δεν περιλαμβάνονταν στην εκπαίδευση ή την ανάπτυξή του. Θα μπορούσε επίσης να διερευνηθεί η δημιουργία ειδικά επιμελημένου συνόλου δεδομένων με υπογραφές που έχουν συλλεχθεί υπό διαφορετικές συνθήκες, με διαφορετικές συσκευές, αναλύσεις και επίπεδα ποιότητας εικόνας.

Δεύτερον η περαιτέρω ανάπτυξη μπορεί να συνδεθεί με την αντικατάσταση του τρέχοντος υβριδικού προσεγγίσης με ένα μοντέλο βαθιάς μάθησης σχεδιασμένο για τη σύγκριση υπογραφών. Αρχιτεκτονικές

όπως τα Siamese Networks, Triplet Networks ή γενικοί αλγόριθμοι μετρικής μάθησης, που μαθαίνουν άμεσα μια αναπαράσταση όπου οι γνήσιες υπογραφές του ίδιου ατόμου είναι κοντά μεταξύ τους και οι πλαστές υπογραφές είναι μακριά στον χώρο χαρακτηριστικών, θα μπορούσαν να ερευνηθούν. Η δημιουργία αυτού του είδους ενός μοντέλου θα απαιτούσε εντελώς διαφορετική διαδικασία εκπαίδευσης, την επιλογή κατάλληλων συναρτήσεων απώλειας και την επαναξιολόγηση των αποτελεσμάτων.

Θα μπορούσε επίσης να εξεταστεί η δημιουργία ενός πλήρως end-to-end μοντέλου, στο οποίο η προεπεξεργασία, η εξαγωγή χαρακτηριστικών και η τελική απόφαση θα αποτελούν ένα ενιαίο εκπαιδευόμενο σύστημα. Σε αυτή την περίπτωση δεν θα απαιτείται ξεχωριστός ταξινομητής ούτε χειροκίνητος συνδυασμός των αποτελεσμάτων με την συνημιτονική ομοιότητα, καθώς το μοντέλο θα εκπαιδεύεται ώστε να προσδιορίζει τον κατάλληλο τρόπο σύγκρισης των δειγμάτων. Η προσέγγιση αυτή θα απαιτούσε σχεδόν πλήρη ανασχεδιασμό του αλγορίθμου ανάλυσης και νέο κύκλο πειραμάτων.

Επιπρόσθετα, η μετάβαση από την offline επαλήθευση βάσει εικόνας στην online επαλήθευση υπογραφών θα μπορούσε να αποτελέσει μελλοντική κατεύθυνση ανάπτυξης της Signalyze. Ένα τέτοιο σύστημα δεν θα περιοριζόταν στην ανάλυση της τελικής εικόνας, αλλά θα λάμβανε υπόψη τη δυναμική διαδικασία δημιουργίας της υπογραφής, όπως την ταχύτητα, την κατεύθυνση των κινήσεων, την πίεση, τη διάρκειά τους και τη χρονική σειρά των διαδρομών της γραφίδας. Για τον σκοπό αυτό θα απαιτούνταν κατάλληλος εξοπλισμός καταγραφής και η ανάπτυξη διαφορετικού μοντέλου ανάλυσης.

Επιπλέον η δημιουργία ενός μηχανισμού συνεχούς επανεκπαίδευσης θα μπορούσε να ληφθεί υπόψη. Καθώς συσσωρεύονται νέα γνήσια δείγματα και νέα αποτελέσματα το σύστημα θα μπορούσε να χρησιμοποιήσει αυτά τα δεδομένα για σταδιακή προσαρμογή της συμπεριφοράς του. Αυτό θα απαιτούσε προσεκτικό σχεδιασμό ώστε να αποφευχθεί η εισαγωγή λανθασμένων δειγμάτων στο σύνολο εκπαίδευσης και να εξασφαλιστεί η αξιοπιστία του μοντέλου.

Επίσης μια πιο εξελιγμένη εξήγηση των αποτελεσμάτων θα μπορούσε να αναπτυχθεί. Αντί της εφαρμογής να παρουσιάζει απλώς το τελικό σκορ και την ετικέτα θα μπορούσε να παρουσιάζει οπτικά τα μέρη της υπογραφής που είχαν τη μεγαλύτερη επιρροή στο αποτέλεσμα, τονίζοντας αυτές τις ενότητες που διαφέρουν σημαντικά από τα δείγματα αναφοράς ή συγκρίνοντας τα συγκεκριμένα δομικά χαρακτηριστικά των χτυπημάτων. Για αυτό το σκοπό θα απαιτούνταν πρόσθετα μοντέλα και αλγόριθμοι εξηγήσιμης τεχνητής νοημοσύνης, καθώς και κατάλληλη οπτικοποίηση στο frontend.

Επιπλέον μια πιο εκτεταμένη προοπτική θα ήταν η δημιουργία μιας εφαρμογής για κινητά. Σε αντίθεση με την τρέχουσα διαδικτυακή εφαρμογή μια εφαρμογή για κινητά θα ήταν σε θέση να αξιοποιήσει καλύτερα τις δυνατότητες της συσκευής, να εκτελέσει μέρος της επεξεργασίας τοπικά και να υποστηρίξει την πραγματική καταγραφή υπογραφών. Θα επέτρεπε επίσης την έρευνα της μηχανικής μάθησης στην συσκευή χωρίς την ανάγκη συνεχούς σύνδεσης στο διαδίκτυο.

Τέλος, πριν από οποιαδήποτε επαγγελματική ή διοικητική χρήση θα απαιτούνταν πολύ πιο εκτεταμένη αξιολόγηση του συστήματος. Η αξιολόγηση αυτή θα περιλάμβανε δοκιμές με πραγματικούς χρήστες, διαφορετικούς τύπους πλαστογράφησης, μεγαλύτερο αριθμό υπογραφόντων και ανεξάρτητα σύνολα δεδομένων. Παράλληλα, θα έπρεπε να αντιμετωπιστούν ζητήματα ασφάλειας, προστασίας προσωπικών δεδομένων και συμμόρφωσης με το ισχύον νομικό πλαίσιο, καθώς οι υπογραφές αποτελούν ευαίσθητα δεδομένα ταυτοποίησης.

Όλες οι αναφερόμενες κατευθύνσεις υπερβαίνουν το πεδίο αυτής της εργασίας καθώς απαιτούν πολύ μεγαλύτερες ποσότητες δεδομένων, υπολογιστικούς πόρους, νέα μοντέλα και διαφορετικές υποδομές. Παρ' όλα αυτά αντιπροσωπεύουν μια φυσική συνέχεια της Signalyze και δείχνουν πώς ένα λειτουργικό

ερευνητικό πρωτότυπο θα μπορούσε να εξελιχθεί στο μέλλον σε ένα πιο ολοκληρωμένο σύστημα επαλήθευσης υπογραφών.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] National Research Council, Biometric Recognition: Challenges and Opportunities. Washington, DC: The National Academies Press, 2010.

[2] National Institute of Standards and Technology (NIST), “Biometrics,” Computer Security Resource Center (CSRC) Glossary.

[3] L. G. Hafemann, R. Sabourin and L. S. Oliveira, “Offline Handwritten Signature Verification — Literature Review,” in Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications (IPTA), pp. 1–8, 2017. DOI: 10.1109/IPTA.2017.8310112.

[4] M. M. Hameed, R. Ahmad, M. L. M. Kiah and G. Murtaza, “Machine learning-based offline signature verification systems: A systematic review,” Signal Processing: Image Communication, vol. 93, Article 116139, 2021. DOI: 10.1016/j.image.2021.116139.

[5] K. S. Radhika and S. Gopika, “Online and Offline Signature Verification: A Combined Approach,” Procedia Computer Science, vol. 46, pp. 1593–1600, 2015. DOI: 10.1016/j.procs.2015.02.089.

[6] K. Roszczewska and E. Niewiadomska-Szynkiewicz, “Online Signature Biometrics for Mobile Devices,” Sensors, vol. 24, no. 11, Article 3524, 2024. DOI: 10.3390/s24113524.

[7] Z. Hashim et al., “A Comparative Study among Handwritten Signature Verification Methods Using Machine Learning Techniques,” Scientific Programming, vol. 2022, Article ID 8170424, 2022. DOI: 10.1155/2022/8170424.

[8] D. Impedovo and G. Pirlo, “Automatic Signature Verification: The State of the Art,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 5, pp. 609–635, 2008. DOI: 10.1109/TSMCC.2008.923866.

[9] Y. Muhtar, W. Kang, A. Rexit, Mahpirat and K. Ubul, “A Survey of Offline Handwritten Signature Verification Based on Deep Learning,” in Proceedings of the 3rd International Conference on Pattern Recognition and Machine Learning (PRML), 2022. DOI: 10.1109/PRML56267.2022.9882188.

- [10] Stergiou K, Ougiaroglou S, Sidiropoulos A. Signature forgery detection using deep and machine learning. *Intelligent Decision Technologies*. 2025;19(3):1429-1444. doi:10.1177/18724981251330068
- [11] L. G. Hafemann, R. Sabourin and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163–176, 2017. DOI: 10.1016/j.patcog.2017.05.012.
- [12] L. Liu, L. Huang, F. Yin and Y. Chen, "Offline signature verification using a region based deep metric learning network," *Pattern Recognition*, vol. 118, Article 108009, 2021. DOI: 10.1016/j.patcog.2021.108009.
- [13] D. Tsourounis, I. Theodorakopoulos, E. N. Zois and G. Economou, "From text to signatures: Knowledge transfer for efficient deep feature learning in offline signature verification," *Expert Systems with Applications*, vol. 189, Article 116136, 2022. DOI: 10.1016/j.eswa.2021.116136.
- [14] W. Xiao and H. Wu, "Learning features for offline handwritten signature verification using spatial transformer network," *Scientific Reports*, vol. 15, Article 9453, 2025. DOI: 10.1038/s41598-025-92704-3.
- [15] W. M. Fatihia, A. Fariza and T. Karlita, "Convolutional neural network enhancement for mobile application of offline handwritten signature verification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 22, no. 4, pp. 931–940, 2024. DOI: 10.12928/TELKOMNIKA.v22i4.25849.
- [16] F. Alonso-Fernandez, J. Fierrez-Aguilar, J. Ortega-Garcia and J. Gonzalez-Rodriguez, "Secure Access System Using Signature Verification over Tablet PC," *IEEE Aerospace and Electronic Systems Magazine*, vol. 22, no. 4, pp. 3–8, 2007. DOI: 10.1109/MAES.2007.351725.
- [17] L. A. B. Cuyos, R. C. E. Jamoner, R. V. S. Siojo and D. R. D. Quiño, "iVERIFY: Signature Recognition and Verification for Android Phones," research/capstone publication.
- [18] R. Reni, Signature Verification Dataset, Kaggle Dataset. Available: <https://www.kaggle.com/datasets/robinreni/signature-verification-dataset>
- [19] CodeIgniter Foundation, CodeIgniter 4 User Guide.
- [20] CodeIgniter Foundation, CodeIgniter Shield Documentation.

- [21] N. Adermann and J. Boggiano, Composer – Dependency Manager for PHP, Composer Documentation.
- [22] D. Hardt, “The OAuth 2.0 Authorization Framework,” IETF RFC 6749, 2012. DOI: 10.17487/RFC6749.
- [23] OpenAI, OpenAI API Documentation – Models and Multimodal Input.
- [24] M. Abadi et al., “TensorFlow: A System for Large-Scale Machine Learning,” in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI), pp. 265–283, 2016.
- [25] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
- [26] G. Huang, Z. Liu, L. van der Maaten and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. DOI: 10.1109/CVPR.2017.243.
- [27] M. Tan and Q. V. Le, “EfficientNetV2: Smaller Models and Faster Training,” in Proceedings of the 38th International Conference on Machine Learning (ICML), vol. 139, pp. 10096–10106, 2021.
- [28] A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv:1704.04861, 2017.
- [29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4510–4520, 2018. DOI: 10.1109/CVPR.2018.00474.
- [30] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” in International Conference on Learning Representations (ICLR), 2015.
- [31] G. Bradski, “The OpenCV Library,” Dr. Dobb’s Journal of Software Tools, 2000.
- [32] Bootstrap Team, Bootstrap 5.3 Documentation.

[33] MAMP GmbH, MAMP & MAMP PRO Documentation. Available at:
<https://documentation.mamp.info/>