

Εξαγωγή υπογραφών από κείμενο και σύγκριση στη βάση δεδομένων

Κωνσταντίνος Σκουρογιάννης

kostskouros@outlook.com

Ηρακλής Παλληκάρης

iraklis.pallikaris@gmail.com

Γενικά

Σε αυτή την εργασία προσπαθούμε να δημιουργήσουμε μια εφαρμογή end-to-end από την αρχή, η οποία θα έχει δύο βασικές λειτουργίες. Η μία λειτουργία είναι εκείνη του αυτόματου εντοπισμού και αποκοπής μιας υπογραφής από έγγραφα και εικόνες, με σκοπό να μπορεί η δεύτερη λειτουργία να την συγκρίνει με άλλες υπογραφές ώστε να εκτιμάται η ομοιότητα των δύο εικόνων και να εκλαμβάνουμε ένα συμπέρασμα ως προς την αυθεντικότητα της υπογραφής. Ταυτόχρονα μια βάση δεδομένων αποθηκεύει και περιέχει αυτά τα δεδομένα ώστε να λειτουργεί μαζί με την εφαρμογή και να έχουν την δυνατότητα ταυτοποίησης υπογραφών βάση χρήστη.

Εισαγωγή

Η υπογραφή είναι ένα από τα πιο δημοφιλή και κοινώς αποδεκτά βιομετρικά χαρακτηριστικά που έχει χρησιμοποιηθεί από την αρχαιότητα για την επαλήθευση διαφορετικών οντοτήτων που σχετίζονται με τον άνθρωπο, π.χ. έγγραφα, έντυπα, τραπεζικές επιταγές, άτομα κ.λπ. Ως εκ τούτου, η επαλήθευση υπογραφών είναι ένα κρίσιμο έργο και έχουν γίνει πολλές προσπάθειες για την άρση της αβεβαιότητας που εμπεριέχει η διαδικασία μη αυτόματης επαλήθευσης ταυτότητας, η οποία καθιστά την επαλήθευση υπογραφής μια σημαντική ερευνητική γραμμή στον τομέα της μηχανικής μάθησης και αναγνώριση προτύπων.

Ανάλογα με τη μορφή εισαγωγής, η επαλήθευση υπογραφής μπορεί να είναι δύο τύπων: (1) online και (2) offline. Η λήψη ηλεκτρονικής υπογραφής χρειάζεται ένα ηλεκτρονική ταμπλέτα μαζί με μια γραφίδα, η οποία μπορεί κυρίως να καταγράψει μια ακολουθία συντεταγμένων της μύτης του ηλεκτρονικού στυλό κατά την υπογραφή. Εκτός από τις συντεταγμένες γραφής της υπογραφής, αυτές οι συσκευές μπορούν επίσης να ανακτήσουν την ταχύτητα γραφής, την πίεση κ.λπ., ως πρόσθετες πληροφορίες, οι οποίες χρησιμοποιούνται στην ηλεκτρονική διαδικασία επαλήθευσης. Από την άλλη πλευρά, η υπογραφή εκτός σύνδεσης καταγράφεται συνήθως από έναν σαρωτή ή οποιονδήποτε άλλο τύπο συσκευών απεικόνισης, οι οποίοι βασικά παράγουν δισδιάστατες εικόνες υπογραφής. Καθώς η επαλήθευση υπογραφής είναι ένα δημοφιλές ερευνητικό θέμα εδώ και δεκαετίες και καταβάλλονται ουσιαστικές προσπάθειες τόσο εκτός σύνδεσης όσο και για σκοπούς επαλήθευσης υπογραφής στο διαδίκτυο. Τα ηλεκτρονικά συστήματα επαλήθευσης γενικά αποδίδουν καλύτερα από τα αντίστοιχα τους εκτός σύνδεσης [3] λόγω της διαθεσιμότητας συμπληρωματικών πληροφοριών όπως σειρά διαδρομής, ταχύτητα γραφής, πίεση κ.λπ. Ωστόσο, αυτή η βελτίωση των επιδόσεων έχει το κόστος της απαίτησης ενός ειδικού υλικού για την καταγραφή της τροχιάς της άκρης του στυλό, την αύξηση του κόστους του συστήματος και τη μείωση των πραγματικών σεναρίων εφαρμογής. Υπάρχουν πολλές περιπτώσεις όπου ο έλεγχος ταυτότητας της υπογραφής εκτός σύνδεσης είναι η μόνη επιλογή, όπως η συναλλαγή επιταγών και η επαλήθευση εγγράφων.

Στην εργασία αυτή εφαρμόζουμε εκτός σύνδεσης μεθόδους που επαληθεύουν μια υπογραφή από έγγραφα και εικόνες που είναι ήδη αποθηκευμένα και έχουν σε κάποιο σημείο που δεν είναι προκαθορισμένο αυτή την υπογραφή.

Πειραματισμός

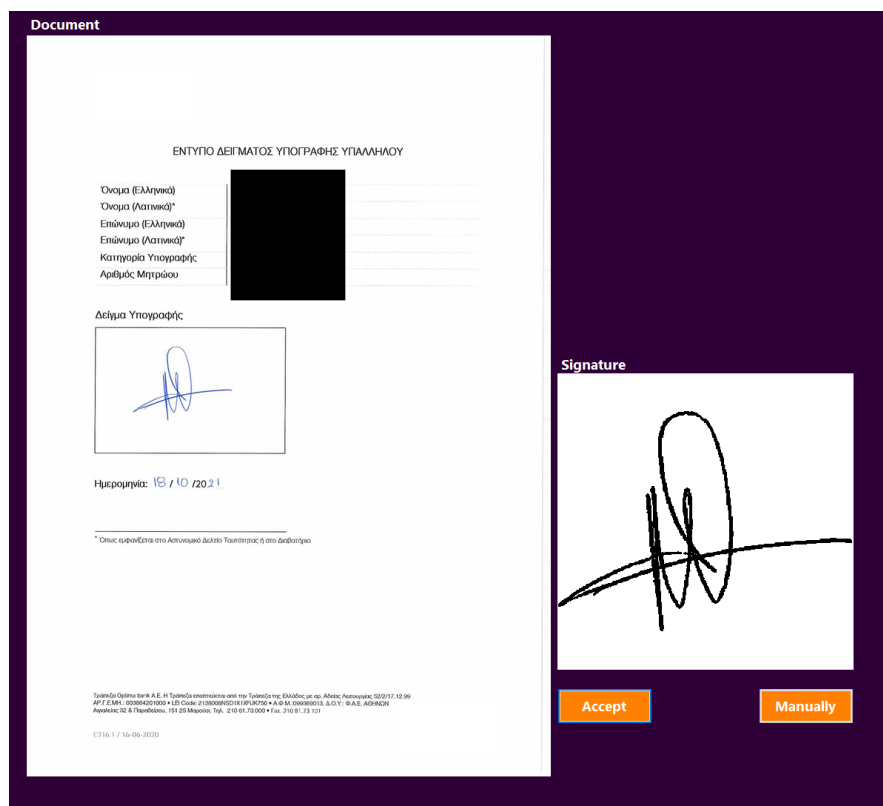
Η υλοποίηση μας είναι ένα σύνολο από scripts σε Python 3.9 όπου έχουν δύο βασικές λειτουργίες:

- 1) Το Extraction, δηλαδή τη δυνατότητα να εξάγεται αυτόματα η υπογραφή από ένα έγγραφο, όπου και αν αυτή βρίσκεται, και να αποθηκεύεται σε βάση δεδομένων αν έχει επιτευχθεί το επιθυμητό αποτέλεσμα.
- 2) Το Comparison, δηλαδή η δυνατότητα να δέχεται η εφαρμογή μια νέα εικόνα από κάποιον χρήστη, η οποία περιέχει μια υπογραφή, και να συγκρίνεται αυτή με υπάρχουσα υπογραφή, ώστε να βρεθεί κατά πόσο μοιάζει και κατ' επέκταση είναι από τον ίδιο χρήστη, ώστε να γίνει ταυτοποίηση.

Συγκεκριμένα, η εφαρμογή υπάρχει σε 2 εκδόσεις, μία που τρέχει απλά στο terminal και μία που τρέχει πίσω από UI που όμως έχει ακριβώς την ίδια λειτουργία. Και οι 2 χρησιμοποιούν μια MySQL database στο background.

UI:

Το θέμα του UI το λύσαμε χρησιμοποιώντας το Visual Studio με το οποίο φτιάξαμε ένα UI σε C# το οποίο φαίνεται στις παρακάτω εικόνες:



Εικόνα 1: Το UI κατά το Extraction

Document Path *

Όνομα *

Επώνυμο *

ΑΦΜ *

Search

Extraction

Εικόνα 2: Το UI κατά την εισαγωγή εικόνας

Comparison By All

Comparison By Name

Signature Extraction

Εικόνα 3: Το UI στο αρχικό μενού

Signature Comparison By All

Document Path *

C:\Users\User\test\test1.pdf

Search

Check

Results

The percentage of similarity is :95% of AFM :000000001

The percentage of similarity is :83% of AFM :000000006

The percentage of similarity is :80% of AFM :000000004

The percentage of similarity is :80% of AFM :000000002

The percentage of similarity is :79% of AFM :000000005

Document Image View

SQL Image View

Document Image View

SQL Image View

Document Image View

SQL Image View

Document Image View

SQL Image View

Document Image View

SQL Image View

Εικόνα 4: Το UI κατά το comparison by db

Βάση Δεδομένων:

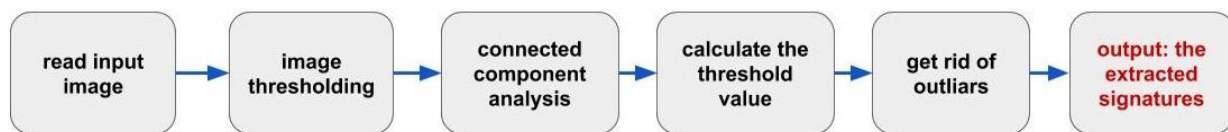
Η βάση δεδομένων στήθηκε τοπικά με την βοήθεια της εφαρμογής xampp και είναι φτιαγμένη σε MySQL. Παρακάτω έχουμε εικόνες που δείχνουν πως εμφανίζονται οι πίνακες:

Table		Action	
<input type="checkbox"/>	signatures	★	Browse Structure Search Insert Empty Drop
		afimi	sig
<input type="checkbox"/>	Edit Copy Delete	102938476	SQL_DB/test1.png
<input type="checkbox"/>	Edit Copy Delete	111111111	SQL_DB/test2.png
<input type="checkbox"/>	Edit Copy Delete	123456789	SQL_DB/test3.png
<input type="checkbox"/>	Edit Copy Delete	574839201	SQL_DB/FormattedSig.png
<input type="checkbox"/>	Edit Copy Delete	675849302	SQL_DB/test5.png
<input type="checkbox"/>	Edit Copy Delete	987654321	SQL_DB/test4.png

Εικόνα 5: Οι πίνακες στη βάση δεδομένων

Extraction:

Η διαδικασία του extraction περιμένει ως είσοδο μια εικόνα (.png ή .jpg) είτε ένα έγγραφο (.pdf) στο οποίο θα γίνει μια διαδικασία, που θα έχει ως έξοδο το αρχείο της υπογραφής. Η διαδικασία του extraction ξεκινάει παίρνοντας το εισαγόμενο αρχείο, που ελέγχεται για το αν είναι ένα από τα παραπάνω είδη αρχείων και αν είναι, αποθηκεύεται ένα αντίγραφο του τοπικά. Έπειτα η πρώτη διεργασία που γίνεται, είναι να αφαιρεθεί το κείμενο της εικόνας το οποίο δεν είναι χειρόγραφο, δηλαδή εκείνο που υπάρχει γραμμένο από κάποιο φωτοτυπικό μηχάνημα ή από υπολογιστή. Έτσι μένει μόνο το κείμενο που είναι χειρόγραφο για να επεξεργαστούμε. Σε αυτό το κείμενο χρησιμοποιούμε την βιβλιοθήκη της python signature-detect 0.1.4. για να εκτελεστεί κατωφλίωση ώστε να βρεθεί η υπογραφή.



Εικόνα 6: Η βασική διαδικασία του extraction

Στη ροή αυτή βλέπουμε πως μετά την κατωφλίωση έρχεται το “connected component analysis” ή αλλιώς επισήμανση συνδεδεμένων στοιχείων. Αυτή η διαδικασία λειτουργεί σε δυαδικές εικόνες ή εικόνες grayscale και είναι δυνατές διαφορετικές μετρήσεις συνδεσιμότητας. Ωστόσο, για τα παρακάτω υποθέτουμε δυαδικές εικόνες εισόδου και συνδεσιμότητα 8-pixel. Ο αλγόριθμος σαρώνει την εικόνα μετακινώντας κατά μήκος μιας σειράς μέχρι να φτάσει σε ένα σημείο p (όπου το p υποδηλώνει το εικονοστοιχείο που θα επισημανθεί σε οποιοδήποτε στάδιο της διαδικασίας σάρωσης) για το οποίο $V=\{1\}$. Όταν αυτό είναι αληθές, εξετάζει τους τέσσερις γείτονες του p που έχουν ήδη συναντηθεί στη σάρωση (δηλαδή τους γείτονες (i) στα αριστερά του p , (ii) πάνω από αυτό και (iii και iv) τους δύο άνω διαγώνιους όρους). Με βάση αυτές τις πληροφορίες, η επισήμανση του p εμφανίζεται ως εξής:

Εάν και οι τέσσερις γείτονες είναι 0, αντιστοιχίσε μια νέα ετικέτα στο p , αλλιώς αν μόνο ένας γείτονας έχει $V=\{1\}$, αντιστοιχίστε την ετικέτα του στο p , αλλιώς εάν περισσότεροι από ένας γείτονες έχουν $V=\{1\}$,

αντιστοιχίστε μία από τις ετικέτες στο p και σημειώστε τις ισοδυναμίες. Μετά την ολοκλήρωση της σάρωσης, τα ισοδύναμα ζεύγη ετικετών ταξινομούνται σε κλάσεις ισοδυναμίας και μια μοναδική ετικέτα εκχωρείται σε κάθε κατηγορία. Ως τελικό βήμα, πραγματοποιείται μια δεύτερη σάρωση μέσω της εικόνας, κατά την οποία κάθε ετικέτα αντικαθίσταται από την ετικέτα που έχει εκχωρηθεί στις τάξεις ισοδυναμίας της. Μετά την διαδικασία αυτή αν αφαιρέσουμε τα outliers μένουν μονάχα οι υπογραφές αν υπάρχουν.

Στη συνέχεια η τοποθετούμε την υπογραφή σε νέα εικόνα μόνη της και επικεντρωμένη κρατώντας όλες τις λεπτομέρειες της, ενώ μετά εφαρμόζουμε μάσκα στην εικόνα μετατρέποντάς την σε ασπρόμαυρη δυαδική εικόνα (χωρίς αποχρώσεις του γκρι).

Τέλος αν ο χρήστης είναι ευχαριστημένος με το αποτέλεσμα μπορεί να κρατήσει την επεξεργασμένη υπογραφή και να εισάγει τον κωδικό του (στην προκειμένη περίπτωση το ονομάσαμε ΑΦΜ μιας και θέλαμε η εφαρμογή να θυμίζει εκείνη μιας τράπεζας για παράδειγμα). Στην περίπτωση που δεν είναι ευχαριστημένος με τα αποτελέσματα μπορεί να αποκόψει την υπογραφή χειροκίνητα με την διεργασία `Manual.py` που έχει την ίδια ακριβώς λειτουργία με την μόνη διαφορά ότι ο χρήστης επιλέγει με το ποντίκι την υπογραφή. Αυτή η λειτουργία υπάρχει γιατί θέλουμε να είμαστε σίγουροι πως στις περιπτώσεις που δεν δουλέψει η αυτόματη επιλογή, η γενική λειτουργία της εφαρμογής θα προχωρήσει κανονικά.

Αποτελέσματα:

Στο δικό μας πειραματισμό δοκιμάσαμε 30 έγγραφα με υπογραφές από τα οποία στα 22 έγινε με επιτυχία η αποκοπή. Το μεγαλύτερο εμπόδιο που φάνηκε να υπάρχει είναι πως μερικές υπογραφές είχαν την μορφή ονόματος οπότε το πρόγραμμα δεν μπορούσε να το ξεχωρίσει από το υπόλοιπο κείμενο εύκολα.

Comparison:

Η δεύτερη λειτουργία που έχει η εφαρμογή είναι η σύγκριση μεταξύ δύο υπογραφών και χωρίζεται σε δύο είδη: Το Comparison by key και το Comparison with db.

Το comparison by key είναι η μέθοδος κατά την οποία η εισαγωγή θα είναι μια εικόνα που περιέχει την υπογραφή που θέλουμε να συγκρίνουμε (μέσα σε κείμενο ή απλή εικόνα), άρα ένα .pdf, .png ή .jpg και η δεύτερη εισαγωγή θα είναι ένα key (στην δική μας περίπτωση το ΑΦΜ) ώστε να βρεθεί στη βάση δεδομένων η αντίστοιχη υπογραφή και να συνεχίσει η διαδικασία. Η έξοδος θα είναι το κατά πόσο αυτή η υπογραφή που εισήχθηκε, ανήκει στον χρήστη ή όχι και κατά πόσο % είναι ο αλγόριθμος σίγουρος. Πρώτα γίνεται formatting στην εικόνα που εισάγουμε και κεντράρεται με την ίδια διαδικασία που ακολουθούμε στο extraction. Έτσι έχουμε μια ασπρόμαυρη Binary εικόνα όπως αυτές της βάσης δεδομένων (ΒΔ). Παρακάτω χρησιμοποιούμε το key για να πάρουμε από την ΒΔ την κατάλληλη εικόνα προς σύγκριση.

Η πρώτη μέθοδος που χρησιμοποιήσαμε είναι σχετικά απλή καθώς δέχεται την εικόνα ως είσοδο και με την βιβλιοθήκη ImageHash 4.2.1 δημιουργείται ένα thumbnail συγκεκριμένων pixel βάση της αρχικής εικόνας και στη συνέχεια συγκρίνεται pixel by pixel για να βγει ένα συνολικό score ομοιότητας.

Η δεύτερη μέθοδος που έδωσε πολύ καλύτερα αποτελέσματα είναι το SSIM (Structural Similarity Index) η οποία είναι μια αντιληπτική μέτρηση που ποσοτικοποιεί την υποβάθμιση της ποιότητας της εικόνας που προκαλείται από επεξεργασία όπως η συμπίεση δεδομένων ή από απώλειες στη μετάδοση δεδομένων. Η διαφορά με άλλες τεχνικές όπως το MSE ή το PSNR είναι ότι αυτές οι προσεγγίσεις εκτιμούν τα απόλυτα σφάλματα. Οι δομικές πληροφορίες είναι η ιδέα ότι τα εικονοστοιχεία έχουν ισχυρές αλληλεξαρτήσεις

ειδικά όταν είναι χωρικά κοντά. Αυτές οι εξαρτήσεις φέρουν σημαντικές πληροφορίες σχετικά με τη δομή των αντικειμένων στην οπτική σκηνή. Η κάλυψη φωτεινότητας είναι ένα φαινόμενο όπου οι παραμορφώσεις της εικόνας (σε αυτό το πλαίσιο) τείνουν να είναι λιγότερο ορατές σε φωτεινές περιοχές, ενώ η κάλυψη αντίθεσης είναι ένα φαινόμενο όπου οι παραμορφώσεις γίνονται λιγότερο ορατές όπου υπάρχει σημαντική δραστηριότητα ή "υφή" στην εικόνα.

Στο Comparison by db η βασική διαφορά είναι πως έχουμε ως είσοδο μόνο την νέα εικόνα με την οποία θα συγκρίνουμε, ενώ η έξοδος θα είναι οι 5 πιο πιθανές υπογραφές από αυτές που είναι στην ΒΔ με το αντίστοιχο σκορ μαζί. Το score προέρχεται από το hash thumbnail ή το SSIM ανάλογα την επιλογή μας.

Αποτελέσματα:

Στη δοκιμή που κάναμε δοκιμάσαμε από τις 30 υπογραφές που είχαμε, 30 νέες από τα ίδια άτομα και 30 που εμείς φτιάξαμε και δοκιμάσαμε να «μοιάζουν» στις αρχικές που είχαμε. Από τις 30 που προήλθαν από τους ίδιους συγγραφείς, τα αποτελέσματα ήταν 25 από τις 30 να βρεθούν όντως ίδιες με SSIM ενώ με την μέθοδο hashmap τα αποτελέσματα ήταν 29 στα 30. Το SSIM κατάφερε να έχει λογικά αποτελέσματα καθώς ο αρχική εικόνα με την τελική έμοιαζε αρκετά σε γενικές γραμμές ενώ είναι και εκείνοι που την υπογραφή τους την έκαναν με λίγο διαφορετικό τρόπο την δεύτερη φορά με αποτέλεσμα να μην το αναγνωρίσει αυτό το SSIM. Από την άλλη το hashmap έκανε μια μεγάλη γενίκευση καθώς αφού μοιάζανε αρκετά μεταξύ τους οι εικόνες, τα thumbnails τους ήταν σχεδόν ίδια. Αυτό το φαινόμενο έγινε ξεκάθαρο στο τεστ των 30 εικόνων που εμείς φτιάξαμε, αφού είχε και εκεί ένα score του ύψους 24 από τα 30, δηλαδή 24 υπογραφές που έγιναν από το δικό μας χέρι τις βρήκε ίδιες με εκείνες από τους αρχικούς συγγραφείς τους, οπότε με σιγουριά μπορούμε να δούμε πως δεν λειτουργεί σωστά. Αυτό είναι επόμενο καθώς το thumbnail δεν είναι αρκετά καλό στο να πιάνει λεπτομέρειες και να αναγνωρίζει σημαντικές διαφορές σε σχεδόν όμοιες εικόνες. Από την άλλη το SSIM είχε αποτελέσματα στο δεύτερο τεστ 12 στα 30 που σίγουρα είναι καλύτερο από το προηγούμενο. Όμως δεν είναι αρκετό καθώς τέτοια συστήματα κρύβουν σημαντικές πληροφορίες συνήθως οπότε δεν θα έπρεπε να επιτρέπεται καμία πλαστογραφημένη υπογραφή να περάσει το τεστ. Αυτό μας λέει λοιπόν ότι τέτοιες τεχνικές που σκοπεύουν να αυθεντικοποιήσουν χρήστες χρειάζονται πιο εξελιγμένες τεχνικές όπως εκπαιδευμένα δίκτυα, τα οποία όμως για να φτιαχτούν χρειάζονται πολλά δεδομένα και hardware ικανό να τα επεξεργαστεί, κάτι που εμείς δεν είχαμε όταν ξεκινήσαμε την εργασία αυτή.

References

Πηγές κώδικα:

Signature Detect: <https://pypi.org/project/signature-detect/>

Πηγές:

Dey, Sounak & Dutta, Anjan & Toledo, J. & Ghosh, Suman & Lladós, Josep & Pal, Umapada. (2017). SigNet: Convolutional Siamese Network for Writer Independent Offline Signature Verification.

R. Plamondon, S. Srihari, Online and off-line handwriting recognition: a comprehensive survey, IEEE TPAMI 22 (1) (2000) 63–84.

D. Impedovo, G. Pirlo, Automatic signature verification: The state of the art, IEEE TSMC 38 (5) (2008) 609–635.

M. K. Kalera, S. N. Srihari, A. Xu, Offline signature verification and identification using distance statistics, IJPRAI 18 (7) (2004) 1339–1360