

Ο ρόλος μου στην εργασία η οποία σχετίζεται με την δημιουργία διεπαφής για την αναγνώριση προσώπων μέσω της βιβλιοθήκης face-recognition της rython και η σύγκριση των διαφόρων μεθόδων που προσφέρει είναι ο ακόλουθος.

Καταρχάς, μάζεψα πληροφορίες για τις τεχνικές βαθιάς μάθησης στις οποίες βασίζεται η βιβλιοθήκη face-recognition οι οποίες είναι ένα support vector machine που παίρνει εικόνες επεξεργασμένες ώστε να γίνουν histograms of oriented gradients και ένα convolutional neural network που προέρχονται από την dlib . Οι πηγές από όπου άντλησα πληροφορίες είναι η ιστοσελίδα όπου εξηγείται η βιβλιοθήκη <https://pypi.org/project/face-recognition/> , τα παραδείγματα για την χρήση της στο GitHub [https://github.com/ageitgey/face\\_recognition/tree/master/examples](https://github.com/ageitgey/face_recognition/tree/master/examples) και το βιβλίο Deep Learning <https://www.deeplearningbook.org/> συγκεκριμένα τα κεφάλαια 5.7.2 όπου εξηγούνται οι support vector machines και 9 όπου εξηγούνται τα convolutional neural networks αν και δεν έχω το απαραίτητο μαθηματικό υπόβαθρο για να τα καταλάβω σε βάθος . Συνολικά αφιέρωσα μία ώρα στην έρευνα.

Στην συνέχεια έφτιαξα τον σκελετό της διεπαφής του προγράμματος χρησιμοποιώντας την βιβλιοθήκη tkinter και τις εντολές για την χρήση των δεδομένων του χρήστη στην αναγνώριση προσώπων ώστε να μπορούν να γίνουν συγκρίσεις ανάλογα με αυτά. Συνολικά αφιέρωσα μία ώρα στην πρώτη έκδοση.

Τέλος, δημιούργησα ένα k-nearest neighbors classifier όπως αυτό στο παράδειγμα [https://github.com/ageitgey/face\\_recognition/blob/master/examples/face\\_recognition\\_knn.py](https://github.com/ageitgey/face_recognition/blob/master/examples/face_recognition_knn.py) για να μπορεί ο χρήστης να αντιστοιχεί το άτομο από μια επιλεγμένη εικόνα στο άτομο που έχει την πλειοψηφία από τους κ κοντινότερους γείτονες του . Συνολικά αφιέρωσα μισή ώρα σε αυτό.