

Περιγραφή Κώδικα: Εφαρμογή Αναγνώρισης Προσώπων με face_recognition και TensorFlow

Ο κώδικας αποτελεί μια εφαρμογή αναγνώρισης προσώπων που χρησιμοποιεί τις βιβλιοθήκες `face_recognition`, `tkinter` και `TensorFlow`. Το πρόγραμμα παρέχει διασύνδεση μέσω του `Tkinter`, επιτρέποντας τη σύγκριση προσώπων από εικόνες και εκτελεί την αναγνώριση προσώπων με χρήση της βιβλιοθήκης `face_recognition`¹ και ενός προσαρμοσμένου μοντέλου CNN που έχει εκπαιδευτεί με το `TensorFlow`².

Αρχικά, ο χρήστης επιλέγει τις παραμέτρους των δύο συγκρίσεων, δηλαδή το μοντέλο που θα χρησιμοποιηθεί, που μπορεί να είναι ένα `Support Vector Machine` που παίρνει εικόνες με μετατρεπμένες με `Histogram Oriented Gradients (hog)` ή ένα `Convolutional Neural Network (cnn)` που προέρχεται από την `dlib` προεκπαιδευμένο. Ακόμα, ο χρήστης επιλέγει αν τα `encoding` που θα πάρει από τις εικόνες θα είναι τα λίγα ή αρκετά περισσότερα σε αριθμό (`small` ή `large`). Μετά, το `tolerance` είναι η απόσταση μεταξύ των `encoding` που θεωρεί αποδεκτή για την σύγκριση και το `num jitters` που παίζει ρόλο στην εύρεση προσώπων που μπορεί να βρισκονται μόνο σε μικρό τμήμα της εικόνας. Έπειτα, ο χρήστης πατάει το κουμπί `Capture Image` αν θέλει να βγάλει μια εικόνα από την κάμερα του υπολογιστή του και να την εποηκεύσει. Επιπλέον, με το κουμπί `Open from file` ο χρήστης μπορεί να επιλέξει εικόνες από τον υπολογιστή του και στο `comptext` εμφανίζεται εαν αναγνωρίστηκαν οι εικόνες, γίνονται συγκρίσεις μεταξύ των δύο, δηλαδή αναφέρονται οι διαφορές στου χρόνους και στην απόσταση. Τέλος, στο `comptext` εμφανίζεται πως αναγνωρίστηκε η εικόνα από το `K-neighbor classifier` και ένα μοντέλο που εκπαιδεύσαμε, και τα δύο με βάση τον φάκελο `train`.

Τα συμπεράσματα που βγάλαμε από τις συγκρίσεις είναι πως η αναγνώριση γίνεται πιο γρήγορα με την παράμετρο `small` παρά την `large` και το `cnn` είναι πολύ πιο αργό από το `hog`. Συμπληρωματικά, ανάλογα με το μέγεθος της εικόνας, επηρεάζεται και ο απαιτούμενος χρόνος. Ακόμα, το `cnn` είναι πιο ακριβές από το `hog` και η ακρίβεια αυξάνεται όταν το `tolerance` είναι κοντά στο 70%. Επιπλέον, το `K-neighbor classifier` έχει μεγαλύτερη ακρίβεια από το μοντέλο μας, καθώς τα `encoding` βασίζονται σε καλύτερα εκπαιδευμένο δίκτυο με πάρα πολύ περισσότερα δεδομένα και περισσότερους πόρους για την εκπαίδευσή του.

Βασικές Λειτουργίες:

Η διαδικασία ανάκτησης εικόνων με το `capture_image` περιλαμβάνει τη χρήση της βιβλιοθήκης `OpenCV (cv2)` για τη λήψη πλαισίων από την κάμερα, ενώ οι εικόνες αποθηκεύονται σε αρχεία για μελλοντική χρήση. Η `self.vid.read()` επιστρέφει δύο τιμές: μια λογική τιμή (`ret`) που υποδεικνύει εάν η λήψη του καρέ ήταν επιτυχής, και το καρέ (`frame`) που ανακτάται από την κάμερα. Εάν η λήψη ήταν επιτυχής (`ret == True`), το καρέ αποθηκεύεται σε ένα αρχείο με μοναδικό όνομα (`captured_image{self.name_counter}.jpg`).

Σύγκριση Προσώπων:

Η σύγκριση προσώπων είναι κρίσιμη για την εφαρμογή, καθώς βασίζεται στην ανίχνευση και τη σύγκριση των χαρακτηριστικών προσώπου ανάμεσα σε δύο εικόνες.

¹ <https://pypi.org/project/face-recognition>

² <https://www.tensorflow.org/tutorials/keras/classification>

Συνάρτηση recognize:

Οι εικόνες φορτώνονται από τα αρχεία που παρέχονται από το χρήστη. Οι περιοχές προσώπου εντοπίζονται στις εικόνες χρησιμοποιώντας το μοντέλο που επιλέγεται από το χρήστη (`model_type`). Τα χαρακτηριστικά προσώπου εξάγονται χρησιμοποιώντας το μοντέλο που επιλέγεται (`encoding_size`) και εκπαιδεύονται με τις γνωστές περιοχές προσώπου. Η σύγκριση προσώπων γίνεται με τη χρήση της `face_recognition.compare_faces()`, ενώ ο υπολογισμός της απόστασης μεταξύ των προσώπων γίνεται με τη χρήση της `face_recognition.api.face_distance()`. Τα αποτελέσματα (αναγνώριση/απόρριψη, απόσταση) και ο χρόνος επεξεργασίας εμφανίζονται στο GUI.

Δημιουργία KNN:

Το μοντέλο KNN Classifier³ χρησιμοποιείται για να προβλέψει την κατηγορία της εικόνας πρόβλεψης. Το μοντέλο KNN Classifier εκπαιδεύεται χρησιμοποιώντας τις εικόνες και του φακέλους ουσιαστικά βρίσκει τα κοντινότερα encodings από τους φακέλους και αναγνωρίζει την εικόνα χρησιμοποιώντας τα ως "ψήφους". Στην διαδικασία, εκτός από την face-recognition, χρησιμοποιούμε και την sklearn για την αποθήκευση του classifier.

Εκπαίδευση του Μοντέλου:

Η διαδικασία εκπαίδευσης του μοντέλου βασίζεται στα δεδομένα του train. Χρησιμοποιείται η διαδικασία του αντίστοιχου tutorial της tensor flow⁴ και στην συνέχεια δημιουργούμε το prediction model για να κάνουμε τις προβλέψεις.

Η Γενική Δομή:

Ο κώδικας αποτελείται από την κλάση app στην οποία δημιουργείται το παράθυρο της Tkinter, για την οποία καλούνται οι `capture_image`, η `compare` και από την οποία καλείται η `find`, η `recognize` και τα αποτελέσματα τυπώνονται στο `comptext`.

Histogram Oriented Gradients (hog)⁵:

Το ιστόγραμμα προσανατολισμένων διαβαθμίσεων (hog) είναι ένας περιγραφέας χαρακτηριστικών που χρησιμοποιείται στην όραση υπολογιστή και στην επεξεργασία εικόνας με σκοπό την ανίχνευση αντικειμένων. Η τεχνική μετράει τα περιστατικά προσανατολισμού διαβάθμισης σε εντοπισμένα τμήματα μιας εικόνας. Αυτή η μέθοδος είναι παρόμοια με αυτή των ιστογραμμάτων προσανατολισμού ακμών, των περιγραφών μετασχηματισμού χαρακτηριστικών αναλλοίωτων σε κλίμακα και των πλαισίων σχήματος, αλλά διαφέρει στο ότι υπολογίζεται σε ένα πυκνό πλέγμα κελιών ομοιόμορφης απόστασης και χρησιμοποιεί επικαλυπτόμενη τοπική κανονικοποίηση αντίθεσης για βελτιωμένη ακρίβεια.

³ <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

⁴ https://www.tensorflow.org/tutorials/load_data/images

⁵ https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

Convolutional Neural Network (cnn)⁶:

Το συνελκτικό νευρωνικό δίκτυο (CNN) είναι ένας κανονικοποιημένος τύπος νευρωνικού δικτύου τροφοδοσίας που μαθαίνει από μόνο του τη μηχανική χαρακτηριστικών μέσω βελτιστοποίησης φίλτρων (ή πυρήνα). Οι διαβαθμίσεις που εξαφανίζονται και οι κλίσεις που εκρήγνυνται, που παρατηρούνται κατά την οπίσθια διάδοση σε προηγούμενα νευρωνικά δίκτυα, αποτρέπονται με τη χρήση ρυθμισμένων βαρών σε λιγότερες συνδέσεις.

⁶ https://en.wikipedia.org/wiki/Convolutional_neural_network