**Project EPL445**

**Tortilla Recognition and Shape Evaluation**

Andeas Thoma - 999999

Gregoris Gregoriadis - 999999

**Εισαγωγή:**

Το Object recognition, με το οποίο και ασχοληθήκαμε σε αυτό το έργο, είναι μια τεχνική computer vision για αναγνώριση αντικειμένων σε εικόνες ή βίντεο και είναι ένα output κλειδί σε deep learning και machine learning αλγόριθμους.

Όταν εμείς οι άνθρωποι βλέπουμε μια εικόνα ή βίντεο, μπορούμε πάρα πολύ εύκολα να αναγνωρίσουμε άτομα, αντικείμενα και άλλες οπτικές λεπτομέρειες. Ο στόχος του object recognition technology είναι να δείξουμε στα computers να κάνουν ότι κάνουμε και εμείς με σκοπό την ανάκτηση ενός επιπέδου κατανόησης του τι περιέχει μια εικόνα.

Το object recognition έχει αξιοσημείωτη συνεισφορά στον τομέα της ασφάλειας, του human-computer interaction, του industrial inspection and automation και άλλα. Χρησιμοποιείται ευρέως σε ζητήματα που αφορούν το computer vision, όπως image annotation, activity recognition, face recognition, face detection και video object segmentation. Επίσης χρησιμοποιείται και στο object tracking σε βίντεο, όπως για παράδειγμα η παρακολούθηση της μπάλας σε ένα παιχνίδι ποδοσφαίρου, ή η παρακολούθηση ενός ατόμου σε βίντεο.

**Μεθοδολογία:**

Το πρώτο μας βήμα ήταν να εκπαιδεύσουμε τον υπολογιστή μας να αναγνωρίζει τορτίγιες. Μπορούμε να το επιτύχουμε αυτό δίνοντας στον υπολογιστή μας 500 τουλάχιστον εικόνες καλής τορτίγιας και 500 τουλάχιστον εικόνες κακής τορτίγιας για εκπαίδευση(training) στην αναγνώριση τορτίγιας, και άλλες 200 τουλάχιστον εικόνες καλής και κακής τορτίγιας για επαλήθευση(testing). Λόγω τον λιγοστών εικόνων που καταφέραμε να βρούμε και την χρήση τους στην εκπαίδευση αναγνώρισης τορτίγιας έχει ως αποτέλεσμα να μην εκπαιδεύεται τελείως σωστά και να μην αναγνωρίζει σε κάποιες περιπτώσεις αν η τορτίγια που του δίνεται είναι καλή ή κακιά.

Έτσι, εξ αιτίας των ανεπαρκή εικόνων για εκπαίδευση του δικτύου για αναγνώριση τορτίγιας με το προαναφερθέν σύστημα, συνεχίσαμε με μια διαφορετική προσέγγιση.

Επιλέξαμε να χρησιμοποιήσουμε ένα υπάρχον σύστημα το οποίο υλοποιεί yolov3 αλγόριθμο για αναγνώριση διάφορων αντικειμένων σε βίντεο και να χρησιμοποιήσουμε αλγόριθμο για shape recognition, ούτως ώστε να μπορούμε να ελέγξουμε ότι μια τορτίγια που αναγνωρίστηκε από το yolov3 είναι κυκλική και συνεπώς έχει το σωστό σχήμα. Έτσι το χρησιμοποιήσαμε αρχικά για να δούμε αν αναγνώριζε τορτίγια. Δυστυχώς, επειδή μια τορτίγια είναι ένα πολύ συγκεκριμένο αντικείμενο, το σύστημά μας αναγνώριζε τις τορτίγιες μας σαν πίτσες. Επειδή όμως το σύστημα που υλοποιήσαμε είναι φτιαγμένο για ένα εργοστάσιο παραγωγής και συσκευασίας τορτίγιας και επειδή μια πίτσα και μια τορτίγια δεν έχουν πολλές διαφορές μεταξύ τους, θεωρήσαμε ότι όποτε το σύστημα αναγνώριζε πίτσα, εμείς θα την αντιμετωπίζαμε σαν τορτίγια.

Αρχικά υλοποιήσαμε ένα GUI με το οποίο ο χρήστης θα επιλέγει ένα βίντεο (αν και το σύστημά μας θα δουλεύει με live camera, για σκοπούς του demo χρησιμοποιούμε υπάρχον βίντεο με τορτίγιες). Στη συνέχεια φορτώνουμε το βίντεο στο σύστημα. Παίρνουμε το βίντεο frame προς frame, το μετατρέπουμε σε black and white και ελέγχουμε αν υπάρχουν white pixels. Αν υπάρχουν white pixels αυτό σημαίνει ότι ξεκίνησε ή περνά ήδη μια τορτίγια από το frame. Αν βρούμε white pixels, τότε ελέγχουμε αν η τορτίγια βρίσκεται στο κέντρο του frame. Αν είναι στο κέντρο, τότε εκτελούμε τον αλγόριθμο του yolov3 για να κάνουμε object recognition και να αναγνωρίσουμε ότι όντος περνάει μια τορτίγια και όχι κάτι άλλο. Μετά την αναγνώριση τορτίγιας, καλούμαστε να αναγνωρίσουμε αν η τορτίγια μας έχει κυκλικό σχήμα. Σε αυτό το σημείο αποφασίσαμε να χρησιμοποιήσουμε την εντολή cv2.HoughCircles η οποία ανιχνεύει και σχεδιάζει κύκλους πάνω σε μια εικόνα. Ακολούθως, ελέγχουμε ότι αναγνωρίστηκε τορτίγια και κύκλος κατά τη διάρκεια που η τορτίγια είναι στο κέντρο του frame μας. Αν αναγνωρίστηκε μόνο τορτίγια και όχι κύκλος, τότε η τορτίγια μας δεν έχει κυκλικό και συνεπώς το σωστό σχήμα και έτσι εμφανίζεται ένα frame το οποίο μας δίχνει την τορτίγια και την χαρακτηρίζει σαν κακής ποιότητας τορτίγια, διαφορετικά εμφανίζει το frame και την χαρακτηρίζει σαν καλής ποιότητας τορτίγια.

Αυτό το σύστημα θα τοποθετηθεί μπροστά από το μηχάνημα συσκευασίας τορτίγιας και μετά από τον χαρακτηρισμό μιας τορτίγιας ώς καλή, τότε το μηχάνημα θα ανοίγει τις πόρτες του για να δεχτεί και να συσκευάσει την τορτίγια. Αν ενημερωθεί από το σύστημα ότι μια τορτίγια είναι κακή, ή αν δεν ενημερωθεί καθόλου τότε οι πόρτες του θα παραμένουν κλειστές με αποτέλεσμα η τορτίγια να απορρίπτεται και να πέφτει έξω από το διάδρομο.

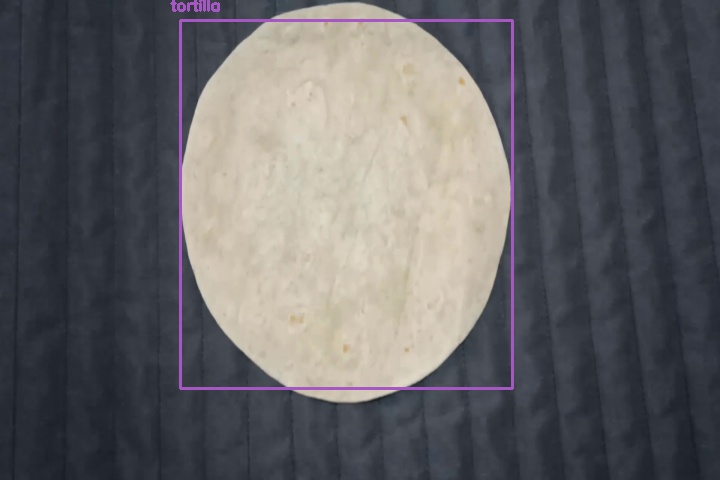
**Αποτελέσματα:**

**Το πρώτο μας παράδειγμα είναι για μια τορτίγια η οποία είναι καλής ποιότητας.**

Η πιο κάτω εικόνα είναι το frame στο οποίο ανιχνεύτηκε μια τορτίγια.



Πιο κάτω φαίνεται και η αναγνώριση της τορτίγιας



Ακολούθως η αναγνώρηση κύκλων

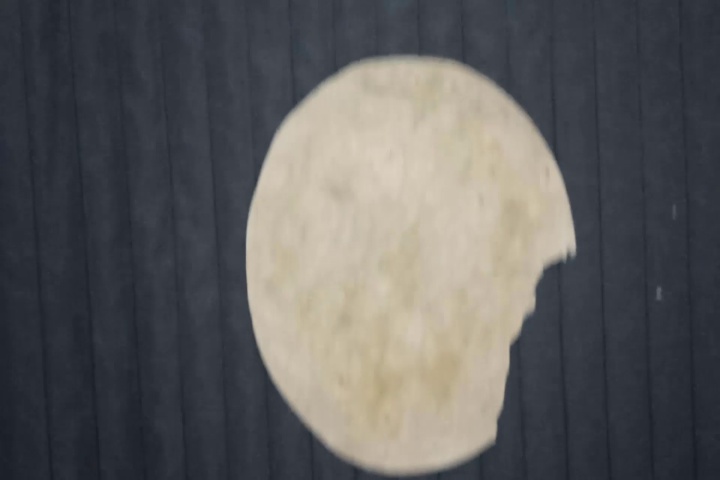


Και αν αναγνωρίστηκε και κύκλος και τορτίγια, τότε εμφανίζεται η πιο κάτω εικόνα με το evaluation ότι η τορτίγια είναι καλής ποιότητας.



**Το παράδειγμα που ακολουθεί είναι για μια κακής ποιότητας τορτίγιας.**

Η πιο κάτω εικόνα είναι το frame στο οποίο ανιχνεύτηκε μια τορτίγια.



Πιο κάτω φαίνεται και η αναγνώριση της τορτίγιας

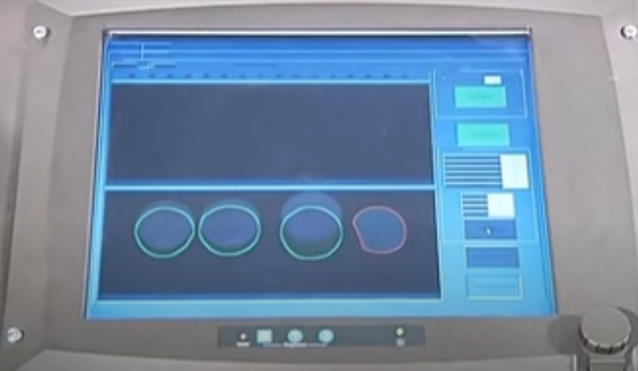
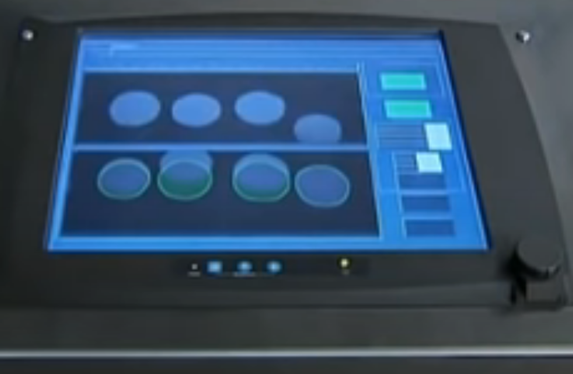


Δεν υπάρχει στη συνέχεια εικόνα αναγνώρισης κύκλων εφόσον η τορτίγια δεν έχει κυκλικό σχήμα και συνεπώς είναι κακής ποιότητας. Έτσι ακολουθεί το επόμενο evaluation:



**Σύγκριση με υπάρχον σύστημα:**

Πιο κάτω φαίνεται ένα σύστημα το οποίο είναι φτιαγμένο για τον ίδιο σκοπό και βλέπουμε ότι δουλεύει σχεδόν με τον ίδιο τρόπο απλά για πολλές τορτίγιεσ ταυτόχρονα.



**Συζήτηση:**

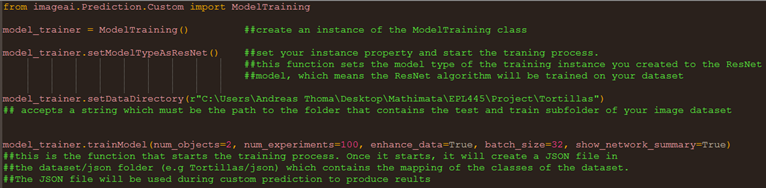
Επιλέξαμε το Object recognition και Shape recognition καθώς είναι αναγκαίο το πρόγραμμά μας να αναγνωρίζει αν αυτό που αξιολογεί είναι τορτίγια ή όχι, διότι το εργοστάσιο που παράγει αυτά τα προϊόντα μπορεί να παράγει και άλλα τρόφημα εκτός από τορτίγιες, και αν αναγνωρίσει τορτίγια τότε με το Shape recognition θα αναγνωρίσει αν η τορτίγια κατέχει τα απαραίτητα κριτήρια μίας σωστής τορτίγιας ώστε να επιτρέψει το πρόγραμμα να περάσει η τορτίγια προς στην συσκευασία της και να μην απορριφθεί, κάτι το οποίο γίνεται πολύ γρήγορα με την κατάληλη εκπαίδευση. Τα μειονεκτήματα όμως είναι πως χρειάζονται πολύ μεγάλο αριθμό δεδομένων (εικόνων καλής και κακής τορτίγιας) ώστε να εκπαιδευτεί πλήρως και επίσης η εκπαίδευση αυτή χρειάζεται πολύ μεγάλο χρονικό διάστημα. Ακόμα ένα μειονέκτημα είναι πως αν η ποιότητα της κάμερας δεν είναι καλή ή η τορτίγια περνά αρκετά γρήγορα από την κάμερα υπάρχει περίπτωση να μην την αναγνωρίσει καθώς οι εικόνες που θα πέρνει το πρόγραμμα μας από το live video θα είναι θολές.

**Πηγές κώδικα και βιβλιογραφία:**

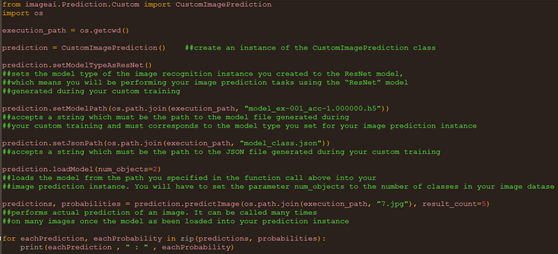
* <https://github.com/OlafenwaMoses/ImageAI/>
* <https://readthedocs.org/projects/imageai/downloads/pdf/latest/>
* <https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html>
* <https://github.com/divikshrivastava/drfoodie/blob/master/main.py>

**Παράρτημα 1 Κώδικας με σχόλια:**

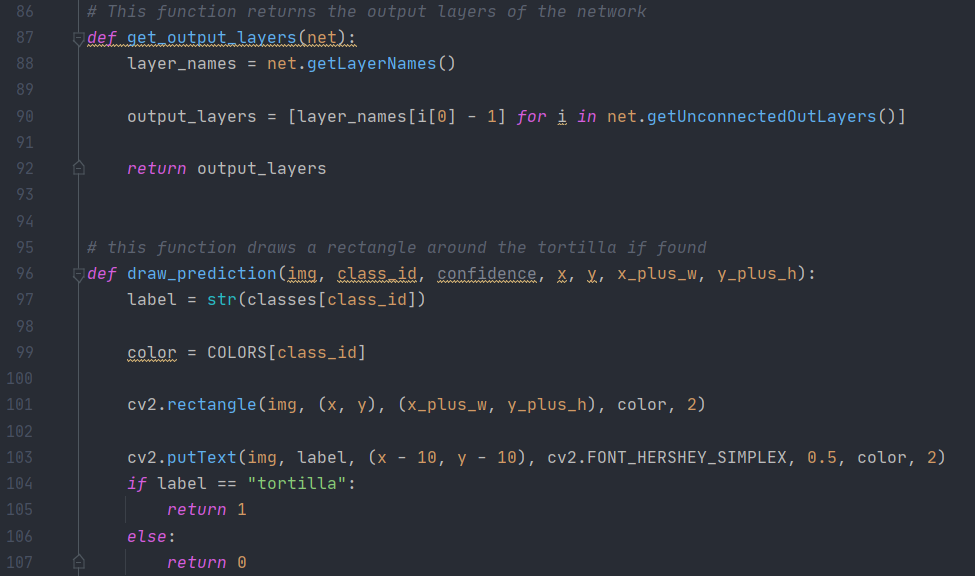
* **Custom Model Training**



* **Custom Detection**

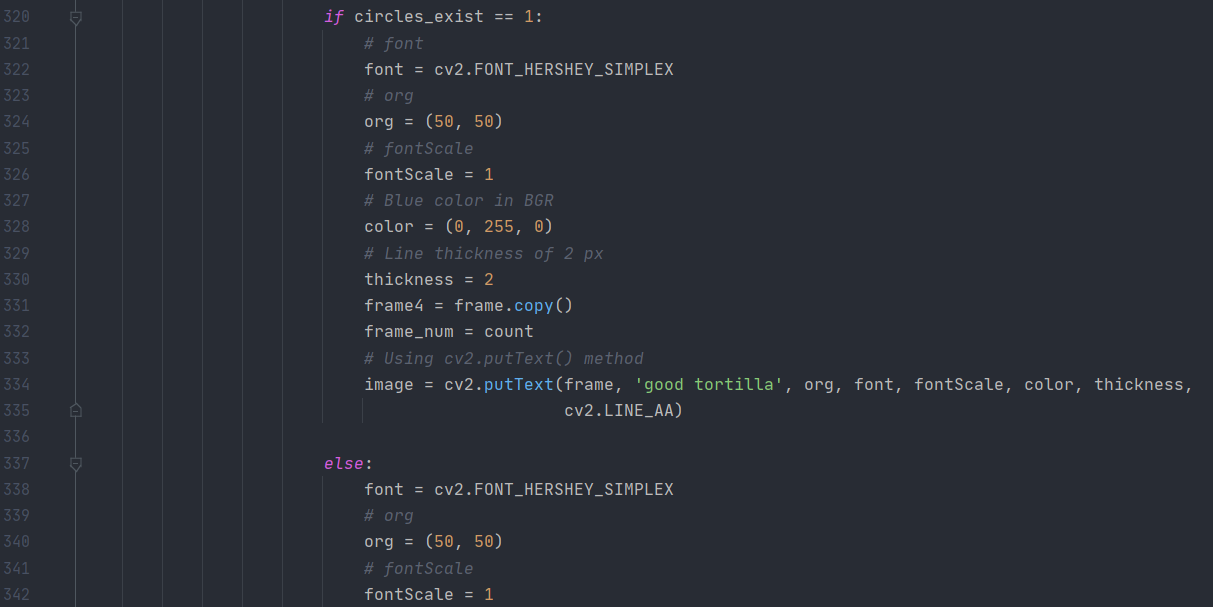
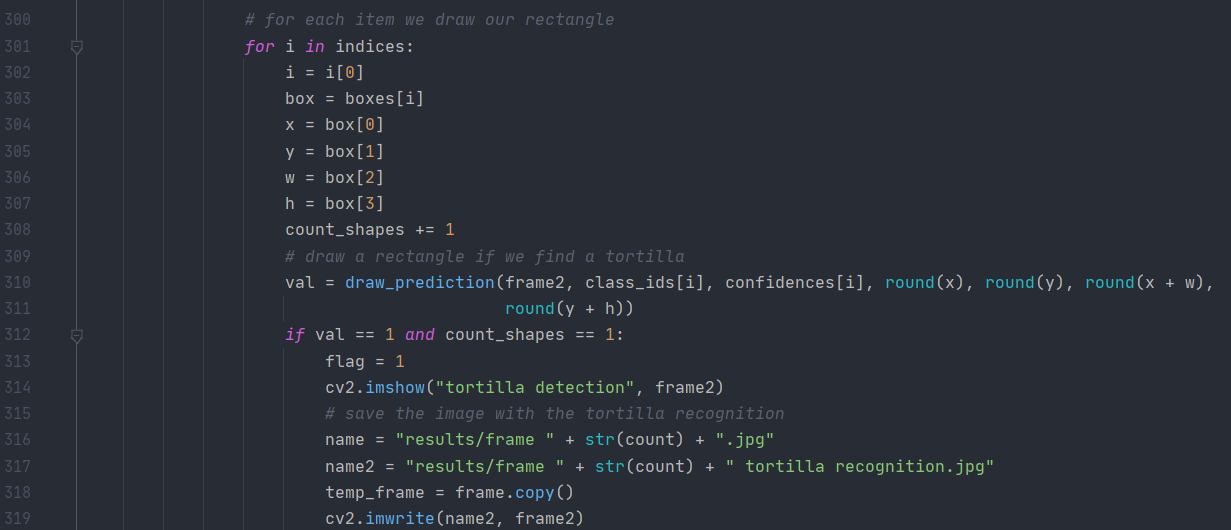
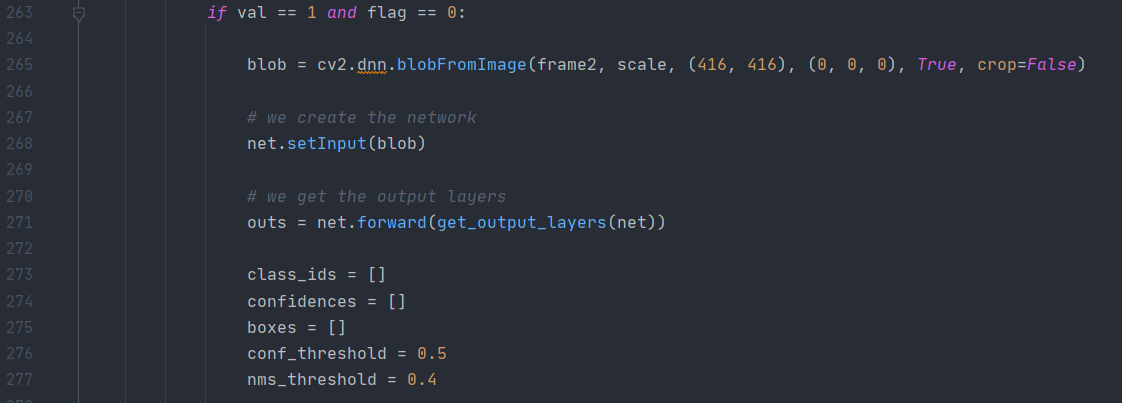
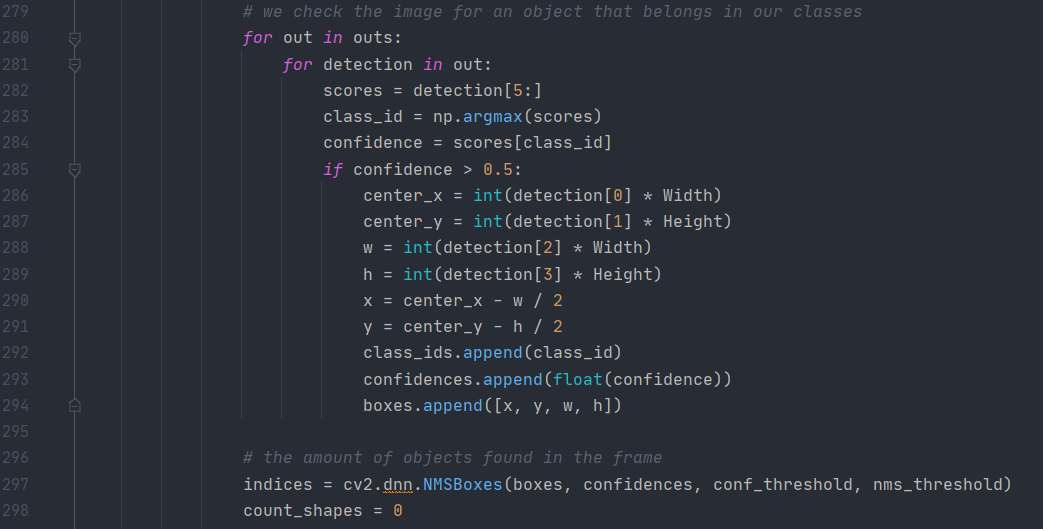
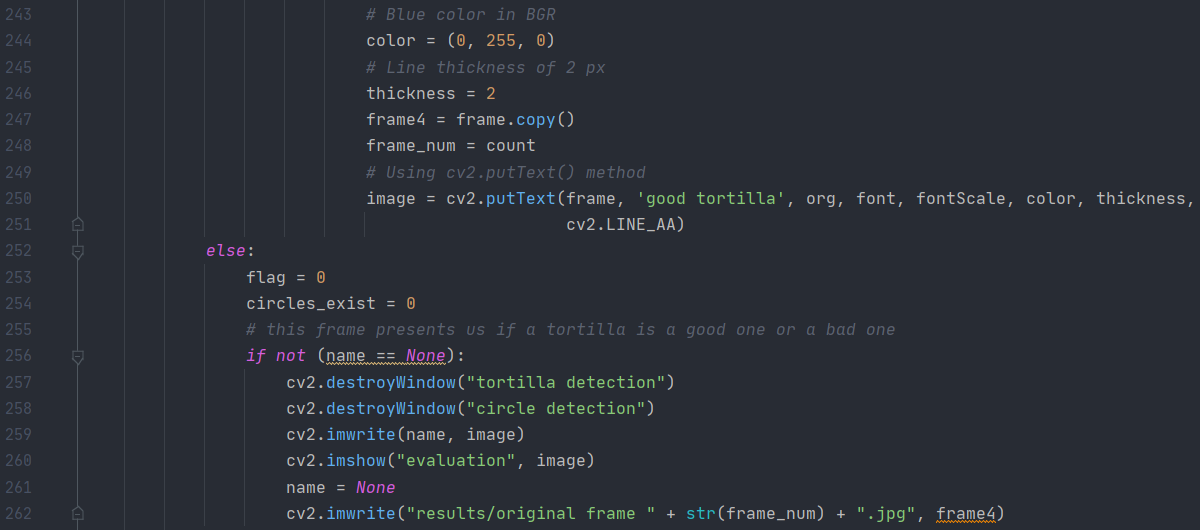
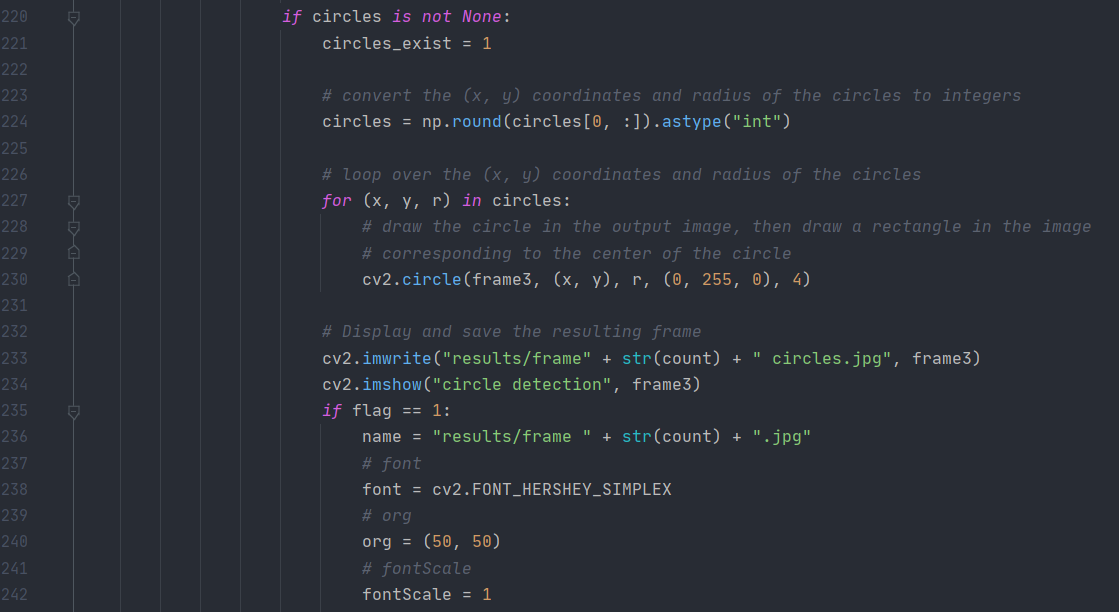
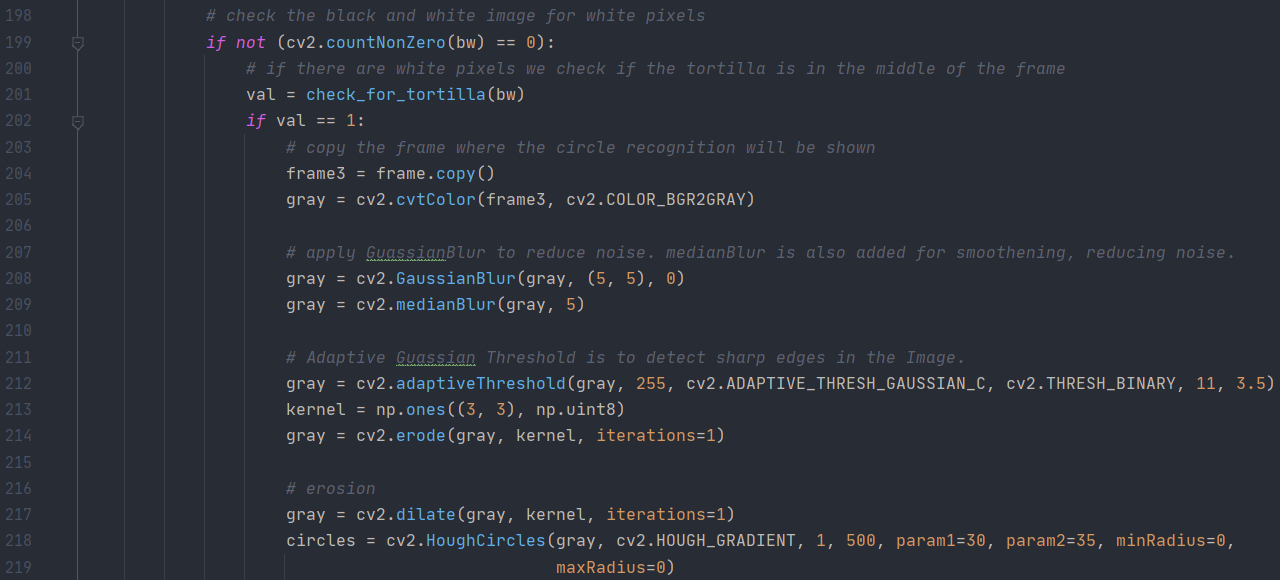


* **Final System**









**Παράρτημα 2 Οδηγίες χρήσης:**

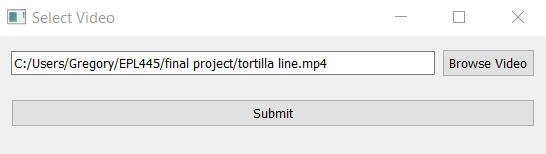
**Οδηγίες Χρήσεις Τελικού Συστήματος:**

Για να τρέξετε το πρόγραμμα δεν θα χρειαστείτε κάποια libraries να είναι εγκατεστημένα.

Πρέπει να έχετε στο ίδιο directory με το πρόγραμμα τα αρχεία yolov3.txt, yolov3.cfg και yolov3.weights που περιέχονται στο .zip.

Επίσης δεν είναι υποχρεωτικό αλλά μπορεί να υπάρχει και ένα subdirectory το οποίο να ονομάζεται results. Εκεί θα αποθηκεύονται οι εικόνες για κάθε αναγνώριση πάνω σε κάθε frame.

Με την εκτέλεση του προγράμματος το πιο κάτω παράθυρο εμφανίζεται και ζητάει από το χρήστη να επιλέξει ένα βίντεο με extensions .mp4, .avi ή .mkv το οποίο θα περιέχει τορτίγιες να έρχονται σε σειρά πάνω σε μια σκούρα επιφάνεια.



**Οδηγίες Χρήσεις Εκπαίδευσης και Αναγνώρισης:**

Requirements για εκπαίδευση και αναγνώριση:

* Python 3.6.2
* Tensorflow 1.13.1
* OpenCV-Python 4.4.0.46
* ImageAI 2.1.5
* Keras 2.2.4
* Numpy 1.16.1

**Εκπαίδευση:**

Στο φάκελο όπου βρίσκεται το “Custom Model Training.py” δημιουργήστε ένα νέο φάκελο Tortillas το οποίο έχει ένα φάκελο train και ένα φάκελο test. Μέσα στο φάκελο train και test να δημιουργήσετε ένα φάκελο bad tortilla και ένα φάκελο good tortilla και γεμίστε αυτούς τους φακέλους με τις ανάλογες εικόνες.

Στον κώδικα συμπληρώστε το path όπου βρίσκονται τα train και test αρχεία  
**model\_trainer.setDataDirectory(r"C:\...\Tortillas")**

**Αναγνώριση:**

Στο φάκελο όπου βρίσκεται το “Custom Detection.py” τοποθετήστε το πιο πρόσφατο .h5 αρχείο που δημιουργήθηκε από την εκπαίδευση το οποίο βρίσκεται στο Tortillas/models.

Στο φάκελο όπου βρίσκεται το “Custom Detection.py” τοποθετήστε το JSON αρχείο που δημιουργήθηκε από την εκπαίδευση το οποίο βρίσκεται στο Tortillas/json.

Στο φάκελο όπου βρίσκεται το “Custom Detection.py” τοποθετήστε μία εικόνα τορτίγιας για αναγνώριση αν είναι καλή ή κακή τορτίγια.

Στον κώδικα συμπληρώστε το όνομα της εικόνας .jpg στην οποία θα προσπαθήσει να αναγνωρίσει το πρόγραμμα

**predictions, probabilities = prediction.predictImage(os.path.join(execution\_path, "ΕΙΚΟΝΑ.jpg"), result\_count=5)**