 ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

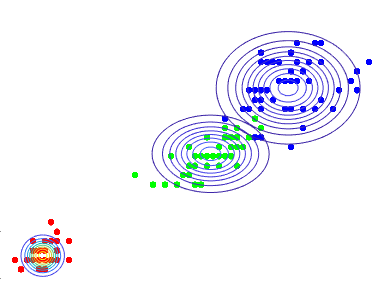
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

Αναγνώριση Προτύπων – Εργασία 1η

***Εξέταση της απόδοσης ταξινομητών (kNN, Bayes, Naive Bayes) και χρήση δικτύων Bayes για συμπερασμό πιθανοτήτων***



*Ον/μο: Δοξαστάκης Γεώργιος*

*Α.Μ.: Μ1613*

*Ον/μο: Μέτσης Γεώργιος*

*Α.Μ.: Μ16--*

Περιεχόμενα

[1. Εισαγωγή 5](#_Toc497383669)

[2. Αρχιτεκτονική συστήματος 7](#_Toc497383670)

[2.1 Στόχοι Σχεδιασμού 7](#_Toc497383671)

[2.2 Περιβάλλον Ανάπτυξης 7](#_Toc497383672)

[2.2.1 IntelliJ IDEA 7](#_Toc497383673)

[2.2.2 JavaFX 8](#_Toc497383674)

[2.2.3 OpenCV 8](#_Toc497383675)

[2.2.4 Apache Commons Math 9](#_Toc497383676)

[2.3 Λογισμικό συστήματος 10](#_Toc497383677)

[2.3.1 Δομή Εφαρμογής 10](#_Toc497383678)

[2.3.2 Thread Περιβάλλοντος Χρήστη (UI Thread) 10](#_Toc497383679)

[2.3.3 Thread Ανάλυσης Βίντεο (Processing Thread) 12](#_Toc497383680)

[2.3.4 Thread Αναπαραγωγής Βίντεο (Playback Thread) 13](#_Toc497383681)

[2.3.5 Αρχεία Καταγραφής 14](#_Toc497383682)

[2.3.6 Ροή Εργασίας 16](#_Toc497383683)

[2.4 Αλγόριθμος Ανάλυσης Πειραμάτων TST 20](#_Toc497383684)

[3. Πειραματικά Αποτελέσματα 23](#_Toc497383685)

[5.1 Πειραματικά Δεδομένα 23](#_Toc497383686)

[5.2 Αποτελέσματα 23](#_Toc497383687)

[6. Συμπεράσματα – Βελτιώσεις 26](#_Toc497383688)

[6.1 Συμπεράσματα 26](#_Toc497383689)

[6.2 Βελτιώσεις 26](#_Toc497383690)

[Βιβλιογραφία 28](#_Toc497383691)

[Παράρτημα A. Κώδικας συστήματος 29](#_Toc497383692)

[1. Main.java 29](#_Toc497383693)

[2. VideoAnalyzer.java 35](#_Toc497383694)

[3. VideoManager.java 41](#_Toc497383695)

[4. layout.fxml 44](#_Toc497383696)

[5. styles.css 47](#_Toc497383697)

[Εικόνα 1: Διαθέσιμες εμπορικές λύσεις EthoVisionXT (πάνω) και MedAssociates TST (κάτω) 6](#_Toc497383648)

[Εικόνα 2: IntelliJ IDEA 7](#_Toc497383649)

[Εικόνα 3: JavaFX 8](#_Toc497383650)

[Εικόνα 4: OpenCV 9](#_Toc497383651)

[Εικόνα 5: Apache Commons Math 9](#_Toc497383652)

[Εικόνα 6: Λειτουργικό διάγραμμα εφαρμογής 10](#_Toc497383653)

[Εικόνα 7: Περιβάλλον χρήστη 11](#_Toc497383654)

[Εικόνα 8: Αρχείο καταγραφής χρονοσειρών 15](#_Toc497383655)

[Εικόνα 9: Αρχείο καταγραφής αποτελεσμάτων 15](#_Toc497383656)

[Εικόνα 10: Μενού επιλογής αρχείου 16](#_Toc497383657)

[Εικόνα 11: Επιλογή περιοχής ανάλυσης 17](#_Toc497383658)

[Εικόνα 12: Ένδειξη προόδου ανάλυσης 18](#_Toc497383659)

[Εικόνα 13: Διαγράμματα αποτελεσμάτων 19](#_Toc497383660)

[Εικόνα 14:. Οπτικοποίηση σκόρ κατά την αναπαραγωγή 20](#_Toc497383661)

[Εικόνα 11: Ανίχνευση οπτικής ροής μέσω του αλγορίθμου Farneback 21](#_Toc497383662)

[Εικόνα 12: Φιλτράρισμα διανυσμάτων (πράσινο: αποδεκτά, κόκκινο: απορριπτέα) 22](#_Toc497383663)

[Εικόνα 13: Δεδομένα κίνησης 22](#_Toc497383664)

[Εικόνα 16: Βίντεο πειραμάτων TST 23](#_Toc497383665)

[Εικόνα 14: Απόδοση αλγορίθμου συναρτήσει των παραμέτρων 24](#_Toc497383666)

[Εικόνα 15: Σύγκριση πραγματικού (πορτοκαλί) με προβλεπόμενο (μπλέ) σκορ 24](#_Toc497383667)

[Εικόνα 17: Διάγραμμα αποτελεσμάτων αυτοματοποιημένου συστήματος (VL) σε σύγκριση με των βαθμολογητών (NK,CD) 25](#_Toc497383668)

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία έχει ως στόχο τον σχεδιασμό ενός αυτοματοποιημένου συστήματος βαθμολόγησης για το συμπεριφορικό πείραμα Tail Suspension Test (TST). Για την βαθμολόγηση του πειράματος χρησιμοποιούνται αλγόριθμοι υπολογιστικής όρασης και σχεδιάζεται ένα γραφικό περιβάλλον με στόχο την απλοποίηση της διαδικασίας και την εξοικονόμηση χρόνου των ερευνητών. Το σύστημα δοκιμάζεται σε πραγματικά πειραματικά δεδομένα και εξετάζεται η απόδοση του σε σύγκριση με τους εμπείρους βαθμολογητές.

ABSTRACT

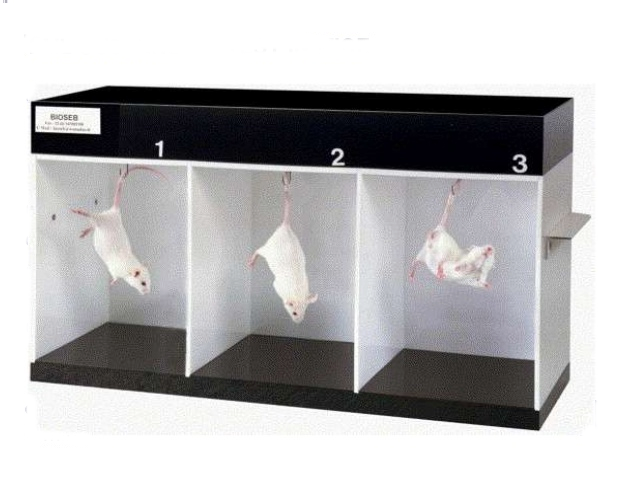
The current dissertation is focused the design of an automated behavioral analysis system for the Tail Suspension Test (TST). The scoring software is designed using computer vision algorithms and is accompanied by a graphical user interface that aims to simplify the process and save time from the user. The system is tested on data from TST experiments and its performance is evaluated using experienced scorers as a reference.

# Εισαγωγή

Κατά την διαλογή νέων φαρμάκων χρησιμοποιούνται διάφορα πειραματόζωα τα οποία είναι στην πλειοψηφία τρωκτικά στα οποία πραγματοποιούνται δοκιμές και αξιολογείται η πιθανή επίδραση των φαρμάκων. Για την μελέτη πιθανών αντικαταθλιπτικών ουσιών πραγματοποιούνται πειράματα σε μικρά τρωκτικά τα οποία θέτονται σε κατάσταση αυξημένου άγχους και παρακολουθείται η διαφορά στις συμπεριφορές που παρουσιάζει ένα πειραματικό group σε σχέση με ένα group αναφοράς. Μια τέτοια μέθοδος είναι το Τail Suspension Test (TST) κατά το οποίο τα πειραματόζωα αναρτώνται μέσω της ουράς σε σημείο από το οποίο δεν μπορούν να φτάσουν κοντινές επιφάνειες για τυπικό διάστημα έξι λεπτών (Can et al., 2011). Κατά την διάρκεια του πειράματος το τρωκτικό εμφανίζει προσπάθειες διαφυγής κουνώντας το σώμα του και σε κάποια διαστήματα οι προσπάθειες παύουν. Ο συνολικός χρόνος κίνησης του πειραματόζωου μεταβάλλεται ανάλογα με την αποτελεσματικότητα των υποψήφιων αντικαταθλιπτικών (Lad et al., 2007).

Για τον υπολογισμό του συνολικού χρόνου κίνησης ένας ερευνητής παρακολουθεί το βίντεο του κάθε πειράματος και βαθμολογεί την κατάσταση κίνησης του πειραματόζωου για κάθε χρονική στιγμή (Can et al., 2011). Η διαδικασία αυτή της βαθμολόγησης πρέπει να πραγματοποιείται με ακρίβεια και από πολλαπλούς βαθμολογητές για κάθε ένα από τα δεκάδες πειραματόζωα του κάθε group. Για την καταγραφή των καταστάσεων κίνησης χρησιμοποιούνται χρονόμετρα ή λογισμικά καταγραφής καταστάσεων μέσω πληκτρολογίου τα οποία δεν ενσωματώνουν την αναπαραγωγή του βίντεο καθιστώντας την διαδικασία δύσκολη στην τυποποίηση και χρονοβόρα.

Για την λύση του προβλήματος βαθμολόγησης υπάρχουν διαθέσιμες στο εμπόριο ολοκληρωμένες λύσεις διεξαγωγής του πειράματος όπως η διάταξη TST της εταιρίας Med Associartes Ιnc. η οποία παρέχει θαλάμους ανάρτησης με ενσωματωμένους αισθητήρες βάρους οι οποίοι με την βοήθεια του συνοδευτικού λογισμικού ανιχνεύουν την κίνηση του πειραματόζωου (Juszczak et al., 2006). Μια άλλη εμπορική λύση είναι το λογισμικό Ethovision XT της εταιρίας Noldus το οποίο πραγματοποιεί συμπεριφορική ανάλυση πολλών τύπων πειραμάτων με χρήση υπολογιστικής όρασης και διατίθεται με τιμή εκκίνησης 5.850 $.



Εικόνα 1: Διαθέσιμες εμπορικές λύσεις EthoVisionXT (πάνω) και MedAssociates TST (κάτω)

Η παρούσα εργασία έχει ως σκοπό την σχεδίαση ενός αυτοματοποιημένου συστήματος βαθμολόγησης πειραμάτων TST μέσω των βίντεο καταγραφής τους το οποίο θα χρησιμοποιεί αλγορίθμους υπολογιστικής όρασης για την εξαγωγή των δεδομένων κίνησης και θα παρέχει στον χρήστη ένα πλήρες περιβάλλον ανάλυσης πειραμάτων. Η χρήση ενός τέτοιου συστήματος απαλλάσσει τον χρήστη από την διαδικασία παρακολούθησης των βίντεο με αποτέλεσμα να μπορεί να χρησιμοποιηθεί μεγαλύτερος αριθμός πειραματόζωων σε κάθε μελέτη.

Παρόμοιες μελέτες έχουν πραγματοποιηθεί για την εξέταση της καταλληλόλητας των διαθέσιμων εμπορικών λύσεων βαθμολόγησης πειραμάτων. Από τα πειράματα TST που πραγματοποιήθηκαν με το σύστημα MedAssociates και το λογισμικό EthoVision XT και συγκρίθηκαν με πραγματικούς βαθμολογητές φάνηκε ότι τα αυτοματοποιημένα συστήματα παρουσιάζουν απόκλιση μερικών μόνο δευτερολέπτων και είναι κατάλληλα για την αξιόπιστη βαθμολόγηση του πειράματος (Juszczak et al., 2006).

# Αρχιτεκτονική συστήματος

## 2.1 Στόχοι Σχεδιασμού

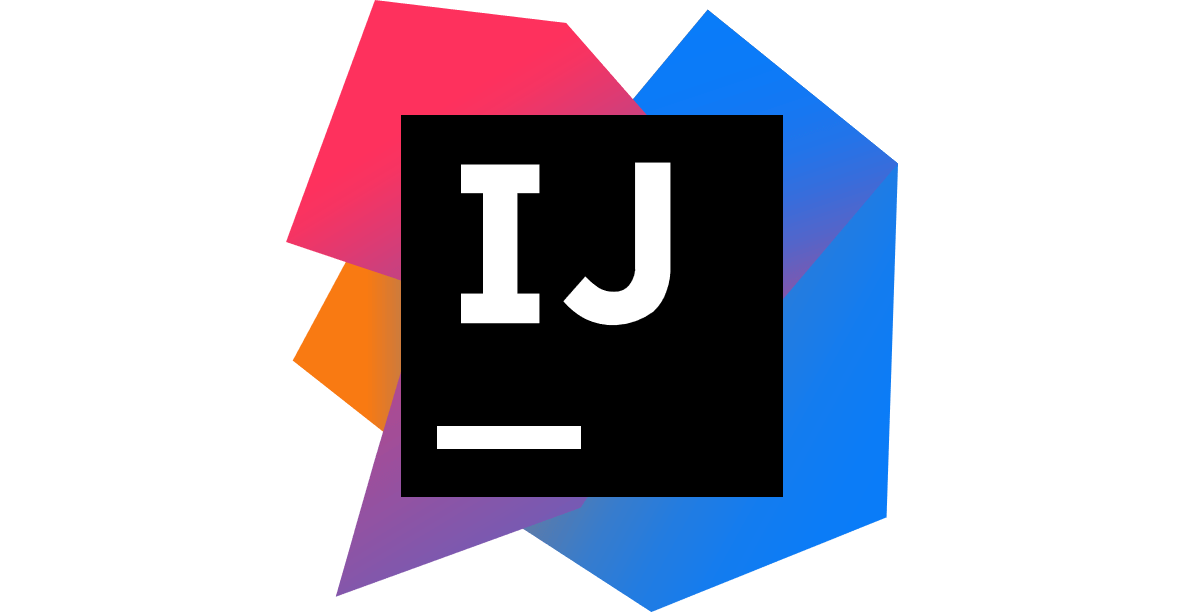
Το σύστημα αυτοματοποιημένης συμπεριφορικής ανάλυσης πειραμάτων με χρήση υπολογιστικής όρασης σχεδιάζεται με τους παρακάτω στόχους.

1. Βαθμολόγηση πειραμάτων TST χωρίς παρουσία χρήστη
2. Έλεγχος διαδικασίας μέσω γραφικού περιβάλλοντος χρήστη (GUI)
3. Κατανοητή ροή εργασίας και εύκολη αποθήκευση αποτελεσμάτων
4. Δυνατότητα αλλαγής παραμέτρων αλγορίθμου ανάλυσης
5. Πλήρης απεικόνιση δεδομένων και οπτικοποίηση κατά την αναπαραγωγή βίντεο πειραμάτων

## Περιβάλλον Ανάπτυξης

### IntelliJ IDEA

Το ολοκληρωμένο περιβάλλον ανάπτυξης IntelliJ IDEA παρέχει την δυνατότητα ανάπτυξης εφαρμογών Java ενσωματώνοντας όλα τα απαραίτητα εργαλεία για κάθε στάδιο δημιουργίας και δοκιμών (JetBrains, 2017). Το IntelliJ IDEA περιέχει δυνατότητες γρήγορης δημιουργίας project, άμεσης εκτέλεσης και διαγνωστικών debugging, διαχείρισης αρχείων και βιβλιοθηκών, δημιουργίας εκτελέσιμων αρχείων, εύκολης μετονομασίας, αυτόματης συμπλήρωσης και έξυπνων προτάσεων βελτιστοποίησης κώδικα και ανίχνευσης λογικών λαθών.



Εικόνα 2: IntelliJ IDEA

Το περιβάλλον επίσης ενσωματώνει διεπαφή με συστήματα version control (github) και υποστηρίξει επεξεργασία συνοδευτικών αρχείων κώδικα (XML,CSS).

### JavaFX

H βιβλιοθήκη JavaFX αποτελεί το σύγχρονο σύστημα σχεδιασμού γραφικού περιβάλλοντος χρήστη (GUI) της Oracle (Docs.oracle.com, 2017). To JavaFX API (Application Programming Interface) περιέχει εύχρηστα αντικείμενα για τον έλεγχο των στοιχείων του γραφικού περιβάλλοντος.



Εικόνα 3: JavaFX

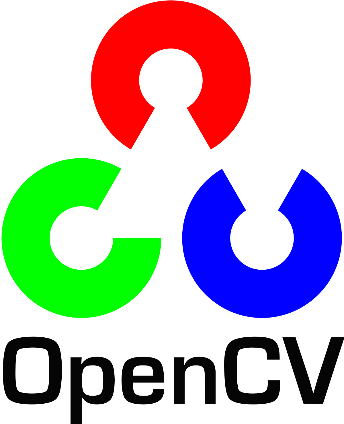
H σχεδίαση του γραφικού περιβάλλοντος πραγματοποιείται σε τροποποιημένη γλώσσα XML και η εμφάνιση των στοιχείων μπορεί να τροποποιηθεί με χρήση αρχείων CSS (Cascading Style Sheets). Στην βιβλιοθήκη JavaFX περιλαμβάνεται επίσης λογισμικό σχεδιασμού και τροποποίησης γραφικού περιβάλλοντος με μεθόδους ‘drag & drop’.

### OpenCV

Η βιβλιοθήκη ανοικτού κώδικα OpenCV αποτελεί μια συλλογή βελτιστοποιημένων αλγορίθμων με εφαρμογή στην υπολογιστική όραση και είναι διαθέσιμη για ακαδημαϊκή και εμπορική χρήση (Opencv.org, 2017). Οι διαθέσιμες συναρτήσεις είναι διαχωρισμένες ανάλογα με το πεδίο εφαρμογής τους ως εξής:

* Βασικές συναρτήσεις διαχείρισης δεδομένων (core)
* Λειτουργίες δημιουργίας γραφικού περιβάλλοντος (highgui)
* Συναρτήσεις επεξεργασίας εικόνας (imgproc)
* Αλγόριθμοι ανάλυσης video (video)
* Μέθοδοι εξαγωγής χωρικών χαρακτηριστικών εικόνας (features2d)
* Αλγόριθμοι ανίχνευσης αντικειμένων (objdetect)
* Λειτουργίες μηχανικής μάθησης (ml)
* Αλγόριθμοι βαθμονόμησης στερεοσκοπικής όρασης (calib3d)
* Συναρτήσεις μετατροπών τύπου δεδομένων (utils)
* Επιπλέον συμβολές χρηστών και πειραματικές λειτουργίες (contrib)

Η βιβλιοθήκη είναι διαθέσιμη στις γλώσσες προγραμματισμού C++, C, Python και Java και υποστηρίζει τα λειτουργικά συστήματα Windows, Linux, Mac OS, iOS και Android.

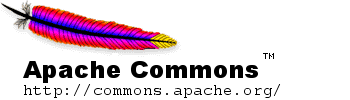


Εικόνα 4: OpenCV

Λόγω του μεγάλου αριθμού χρηστών της βιβλιοθήκης υπάρχουν διαθέσιμα παραδείγματα και υποστήριξη σε εξειδικευμένα θέματα υπολογιστικής όρασης.

### Apache Commons Math

Η βιβλιοθήκη Apache Commons Math (Commons.apache.org, 2017) είναι μια μικρού μεγέθους βιβλιοθήκη η οποία περιέχει όλες τις απαραίτητες λειτουργίες μαθηματικών και στατιστικής σε εύχρηστη μορφή για ανάπτυξη εφαρμογών.



Εικόνα 5: Apache Commons Math

## Λογισμικό συστήματος

### Δομή Εφαρμογής

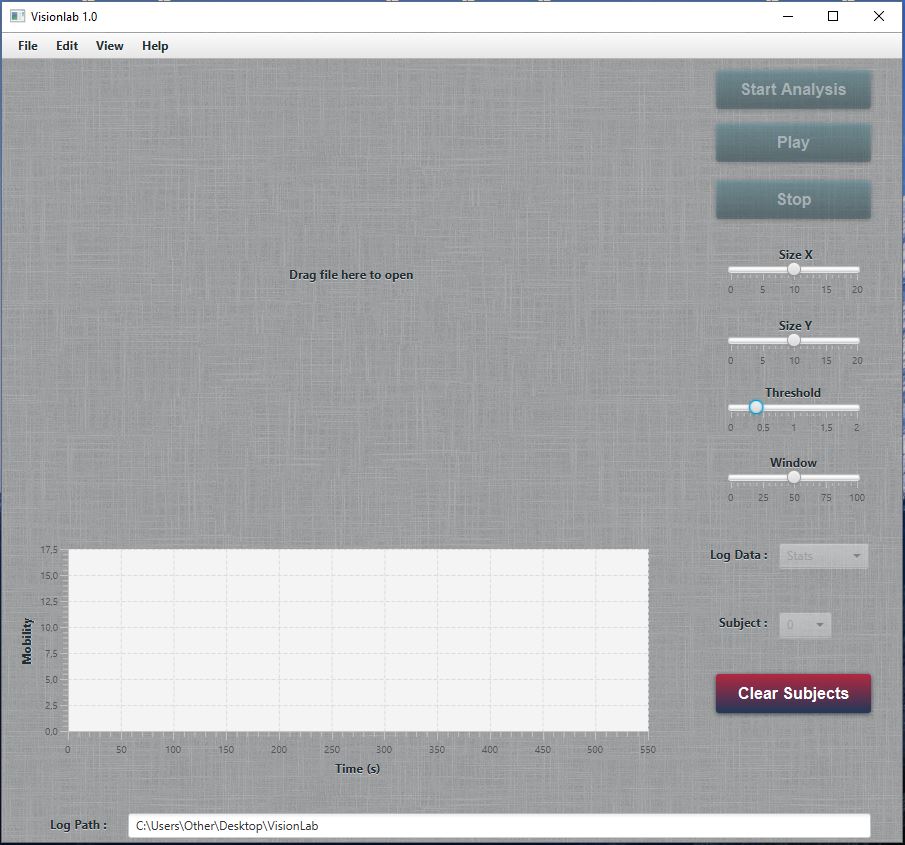
Η εφαρμογή υλοποιείται ως ένα εκτελέσιμο πρόγραμμα σε μορφή JAR, ένα εκτελέσιμο αρχείο Java που περιλαμβάνει τον εκτελέσιμο κώδικα και τα απαραίτητα συνοδευτικά αρχεία. Η εφαρμογή βασίζεται στο γραφικό περιβάλλον και περιλαμβάνει δυο επιπλέον νήματα (threads) εκτέλεσης τα οποία εκτελούν τα υπολογιστικά απαιτητικά μέρη της εφαρμογής. Το κυρίως thread διαχειρίζεται το περιβάλλον χρήστη ενώ τα αλλά δυο διαχειρίζονται την αναπαραγωγή και την ανάλυση των πειραμάτων (Εικόνα 6).



Εικόνα 6: Λειτουργικό διάγραμμα εφαρμογής

### Thread Περιβάλλοντος Χρήστη (UI Thread)

Το thread του περιβάλλοντος χρήστη είναι το κύριο thread της εφαρμογής και αναλαμβάνει την διαχείριση των γραφικών στοιχείων ελέγχου όπως η απόκριση σε click κουμπιών, αλλαγής επιλογών ή αλλαγής θέσης συρόμενων ρυθμιστών. Στο ίδιο thread επίσης εκτελούνται οι απαραίτητες μετατροπές για την απεικόνιση δεδομένων (εικόνες και διαγράμματα) τα οποία προέρχονται από τα thread επεξεργασίας και αναπαραγωγής.



Εικόνα 7: Περιβάλλον χρήστη

Το γραφικό περιβάλλον (Εικόνα 13) αποτελείται από τον χώρο απεικόνισης του βίντεο στον οποίο ο χρήστης μπορεί να εισάγει αρχεία μέσω drag & drop, από τα πλήκτρα έναρξης/τερματισμού ανάλυσης και αναπαραγωγής, τις κυλιόμενες ρυθμίσεις των παραμέτρων ανάλυσης και τον χώρο απεικόνισης διαγραμμάτων.

Στο αντικείμενο του UI thread περιλαμβάνονται οι απαραίτητες μεταβλητές για τον έλεγχο κατάστασης της εφαρμογή οποίες περιέχουν πληροφορία για ανοικτά αρχεία, τρέχουσες ενέργειες και επιτρεπτές δράσεις.

Το UI Thread αναλαμβάνει την διαχείριση των εξής γεγονότων που δημιουργούνται από το γραφικό περιβάλλον και από τα υπόλοιπα threads:

* Γραφικό περιβάλλον (GUI)
  + onStartAnalysis – Εκκίνηση ανάλυσης από το αντίστοιχο κουμπί
  + onPlay – Εκκίνηση αναπαραγωγής από το αντίστοιχο κουμπί
  + onStop - Λήξη αναπαραγωγής από το αντίστοιχο κουμπί
  + onMouseClick – Απόκριση σε κλικ για επιλογή περιοχών ανάλυσης
  + onFileEnter - Έλεγχος καταλληλόλητας αρχείου όταν σύρεται στην περιοχή
  + onFileDrop – Άνοιγμα αρχείου κατά την εναπόθεση στην περιοχή
  + onFileOpen – Ανοιγμα παράθυρου επιλογής αρχείου
  + onFileClose -Κλείσιμο τρέχοντος αρχείου
  + onClearSubjects – Εκκαθάριση των επιλεγμένων περιοχών ανάλυσης μέσω του αντίστοιχου κουμπιού
  + onPlotDataChanged – Απεικόνιση διαγραμμάτων για το επιλεγμένο πειραματόζωο
* Thread Αναπαραγωγής
  + onImage – Απεικόνιση εισερχόμενου frame στο γραφικό περιβάλλον

### Thread Ανάλυσης Βίντεο (Processing Thread)

Στο thread ανάλυσης βίντεο εκτελείται ο αλγόριθμος ανάλυσης του πειράματος καθώς και λες οι σχετικές διαδικασίες όπως η προσπέλαση του αρχείου βίντεο και η προετοιμασία των δεδομένων για τα διαγράμματα απεικόνισης.

Το thread ανάλυσης έχει πρόσβαση στα δεδομένα του UI Thread τα οποία περιλαμβάνουν το όνομα του αρχείου προς ανάλυση, τις επιλεγμένες περιοχές ανάλυσης, τις ρυθμίσεις του χρήστη και την τοποθεσία αποθήκευσης των αρχείων καταγραφής.

Κατά την ανάλυση λαμβάνονται διαδοχικά frames από το αρχείο του πειράματος και αναλύονται ξεχωριστά για κάθε περιοχή ενδιαφέροντος, στην συνέχεια τα δεδομένα της ανάλυσης αποθηκεύονται στις κατάλληλες δομές για την απεικόνιση τους και εξάγονται σε αρχεία για μελλοντική χρήση.

Αναλυτικά η διαδικασία της ανάλυσης περιλαμβάνει τα εξής βήματα:

* Άνοιγμα του αρχείου βίντεο
* Αρχικοποίηση προηγούμενου frame με το πρώτο frame του βίντεο
* Για κάθε διαθέσιμο πειραματόζωο
  + Δημιουργία πινάκων αποθήκευσης χρονοσειρών ανάλυσης
* Επανάληψη για κάθε frame του βίντεο:
  + Λήψη του frame από το βίντεο
  + Για κάθε διαθέσιμο πειραματόζωο
    - Αποκοπή περιοχής ενδιαφέροντος και κλιμάκωση σε σταθερό μέγεθος
    - Υπολογισμός της οπτικής ροής σε σχέση με το προηγούμενο frame
    - Φιλτράρισμα διανυσμάτων και αποθήκευση του συνολικού αριθμού τους και του συνόλου των μέτρων τους
  + Αντικατάσταση του προηγουμένου frame με το τωρινό
  + Αποστολή κατάστασης προόδου στο UI Thread
* Κλείσιμο του αρχείου βίντεο
* Για κάθε διαθέσιμο πειραματόζωο
  + Επεξεργασία χρονοσειρών
  + Βαθμολόγηση καταστάσεων
* Αποστολή χρνοσειρών στο UI Thread
* Αποθήκευση αρχείων καταγραφής
* Παύση εκτέλεσης του thread

### Thread Αναπαραγωγής Βίντεο (Playback Thread)

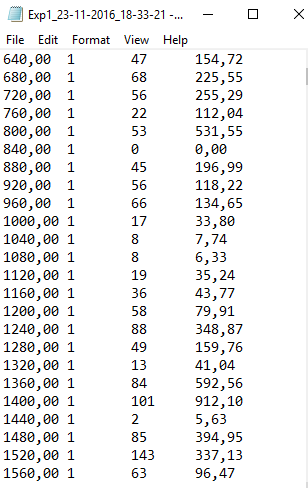
Το thread αναπαραγωγής αναλαμβάνει την μεταφορά καθενός frame εικόνας από το αρχείο στο περιβάλλον χρήστη και εξασφαλίζει την αναπαραγωγή σε πραγματικό χρόνο. Εφόσον έχει προηγηθεί ανάλυση του βίντεο δίνεται η επιλογή απεικόνισης της βαθμολόγησης κίνησης κατά την αναπαραγωγή μέσω της αλλαγής του χρωματισμού του πλαισίου της περιοχής ενδιαφέροντος.

Αναλυτικά το thread αναπαραγωγής εκτελεί την εξής διαδικασία:

* Άνοιγμα του αρχείου βίντεο
* Επανάληψη για κάθε frame του βίντεο:
  + Λήψη του frame από το βίντεο
  + Σχεδίαση των περιοχών ανάλυσης στην εικόνα
  + Επικάλυψη της πληροφορίας βαθμολόγησης στα περιγράμματα (εφόσον έχει πραγματοποιηθεί ανάλυση)
  + Καθυστέρηση για την τήρηση του ρυθμού αναπαραγωγής
  + Αποστολή του frame στο UI Thread προς εμφάνιση
* Κλείσιμο του αρχείου βίντεο
* Παύση εκτέλεσης του thread

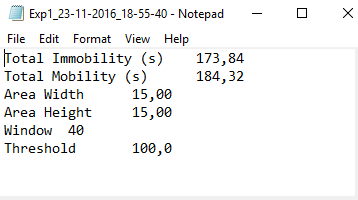
### Αρχεία Καταγραφής

Η εφαρμογή δίνει στον χρήστη την δυνατότητα αποθήκευσης των δεδομένων ανάλυσης σε δυο μορφές αρχείου κειμένου. Στον πρώτο τύπο αρχείου καταγραφής (Εικόνα 8) αποθηκεύονται τα δεδομένα κίνησης των πειραματόζωων ως χρονοσειρές για επιπλέον επεξεργασία εκτός της εφαρμογής.



Εικόνα 8: Αρχείο καταγραφής χρονοσειρών

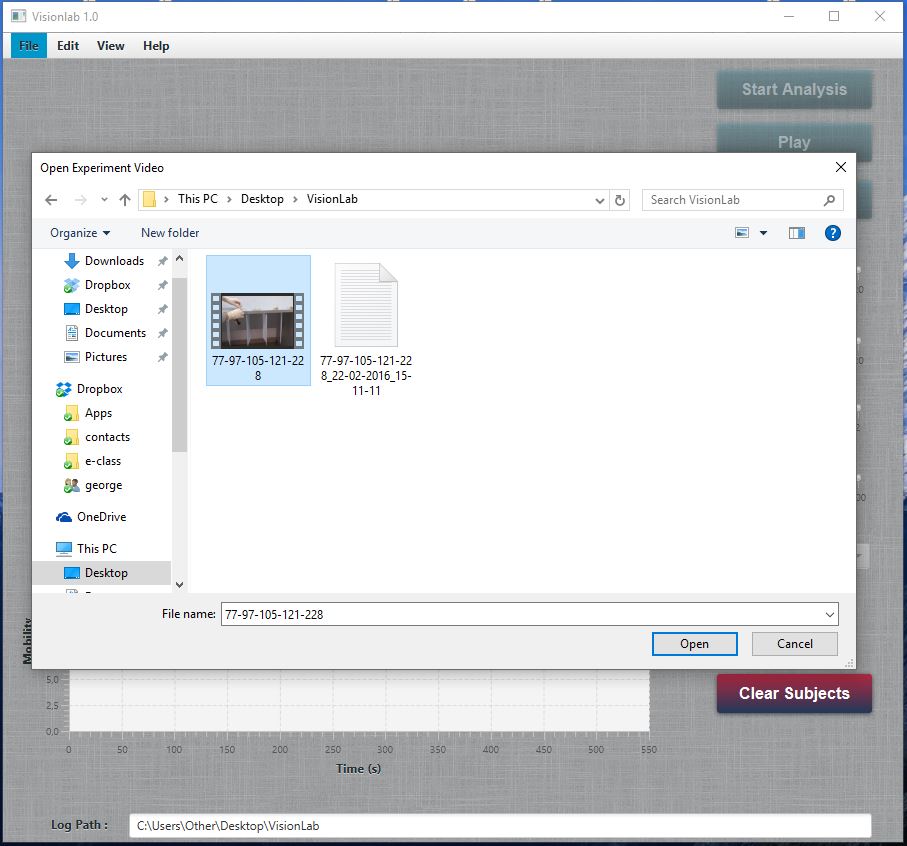
Στον δεύτερο τύπο αρχείου (Εικόνα 8) αποθηκεύονται μόνο οι τελικές πληροφορίες βαθμολόγησης καθώς και οι ρυθμίσεις με τις οποίες πραγματοποιήθηκε η ανάλυση.



Εικόνα 9: Αρχείο καταγραφής αποτελεσμάτων

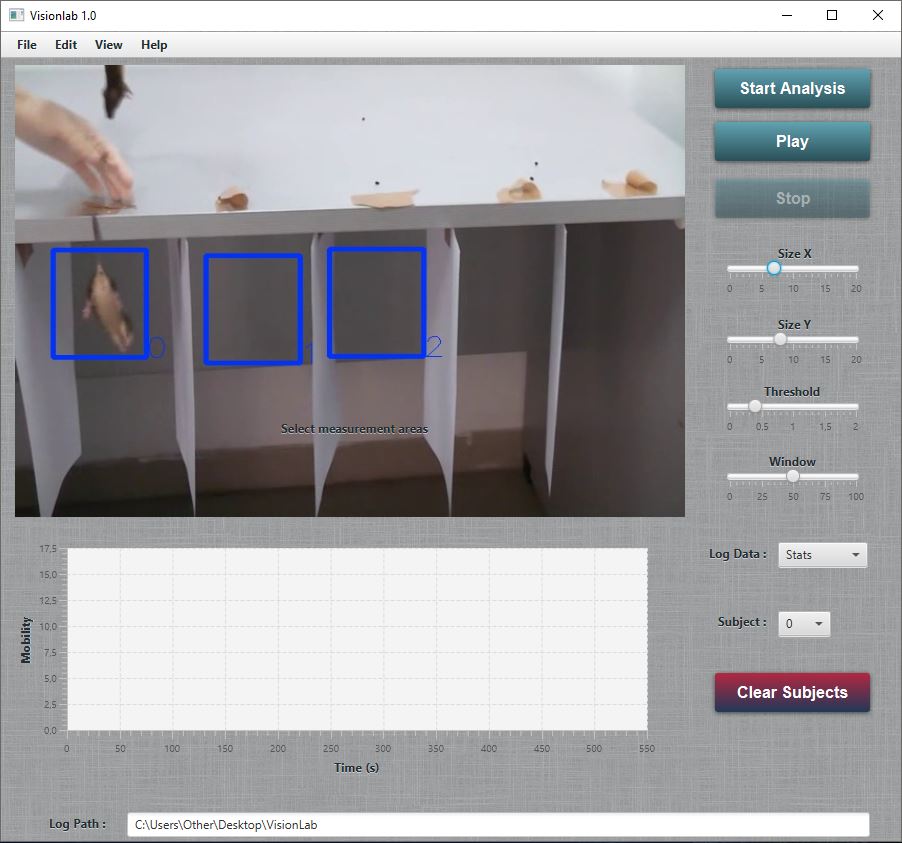
### Ροή Εργασίας

Η εφαρμογή προσφέρει στον χρήστη ένα απλό περιβάλλον με κατανοητές ενδείξεις ως προς τον τρόπο χρήσης. Για την ανάλυση ενός βίντεο ο χρήστης εισάγει το βίντεο του πειράματος είτε μέσω drag & drop είτε μέσω του μενού File->Open (Εικόνα 10).



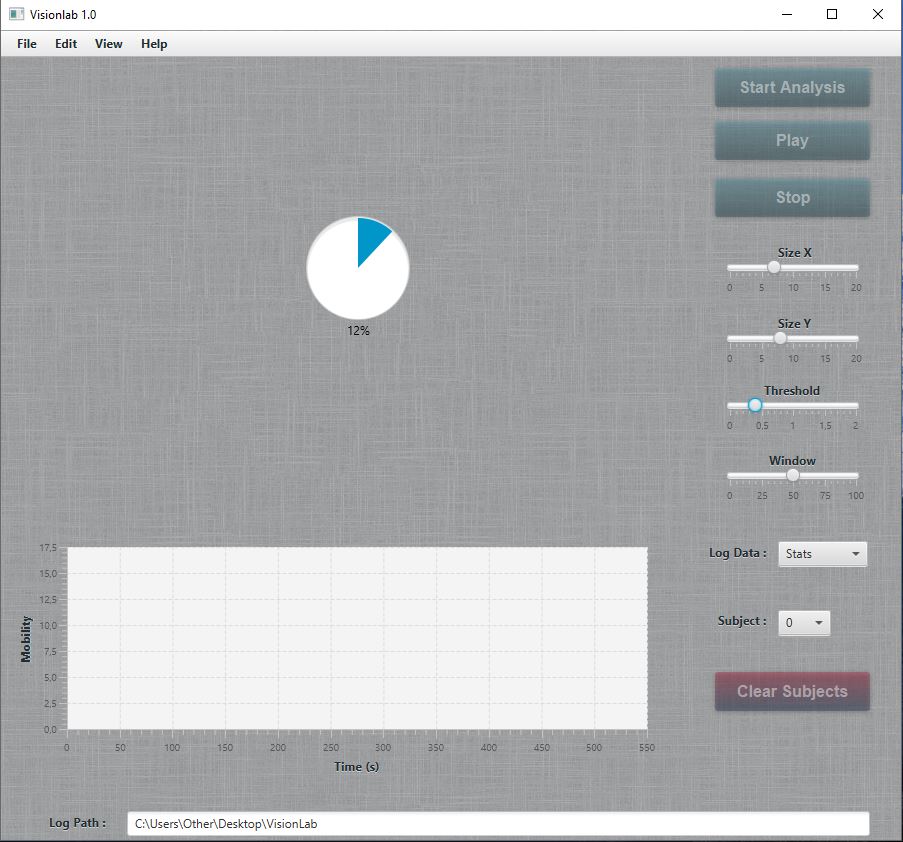
Εικόνα 10: Μενού επιλογής αρχείου

Στην συνέχεια ο χρήστης καλείται να επιλέξει τις περιοχές στις οποίες πραγματοποιείται η ανάλυση (μια για κάθε πειραματόζωο) (Εικόνα 11.). Ο χρήστης μπορεί να προσαρμόσει το μέγεθος της περιοχής με τους ρυθμιστές Size X και Size Y έχοντας υπ’ όψιν ότι η περιοχή πρέπει να περιλαμβάνει το πειραματόζωο και να υπάρχει περιθώριο για τις κινήσεις του. Αν ο χρήστης σε αυτό το στάδιο πραγματοποιήσει μια λάθος επιλογή μπορεί να επαναλάβει την διαδικασία πατώντας το κουμπί Clear Subjects.



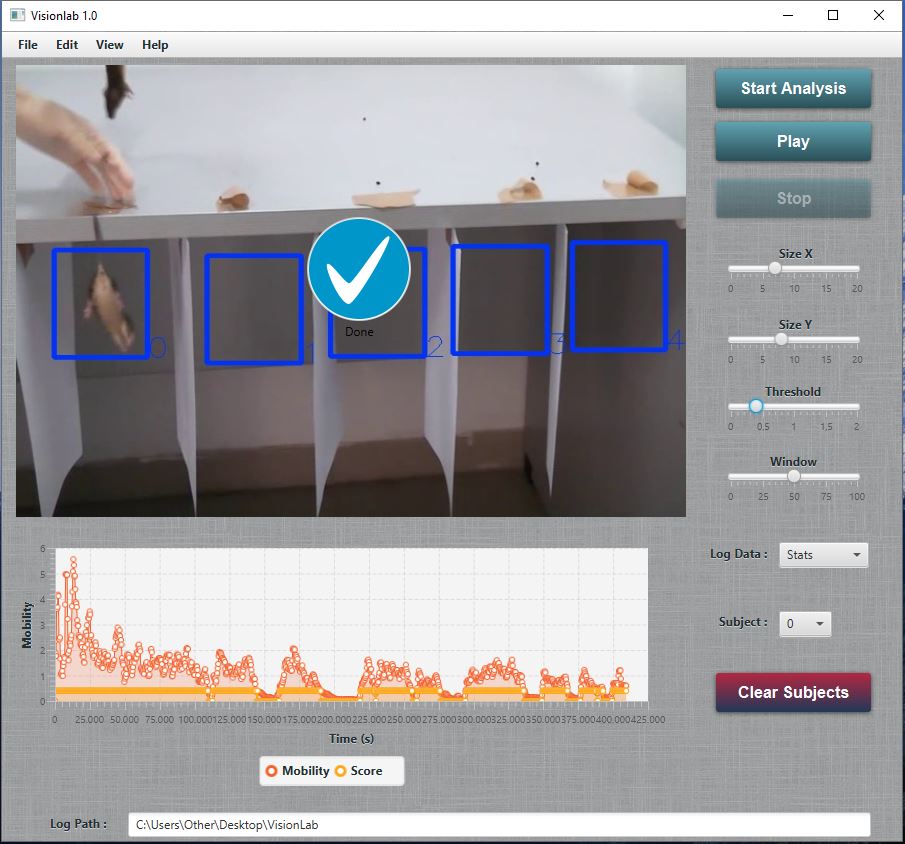
Εικόνα 11: Επιλογή περιοχής ανάλυσης

Αφού έχουν επιλεγεί οι περιοχές ανάλυσης και έχουν καθοριστεί οι παράμετροι του αλγορίθμου βαθμολόγησης ο χρήστης μπορεί να ξεκινήσει την ανάλυση του βίντεο μέσω του κουμπιού Start Analysis. Κατά την διάρκεια της ανάλυσης στην θέση της προεπισκόπησης του βίντεο προβάλλεται γραφική ένδειξη της προόδου όπως φαίνεται στην εικόνα 12.



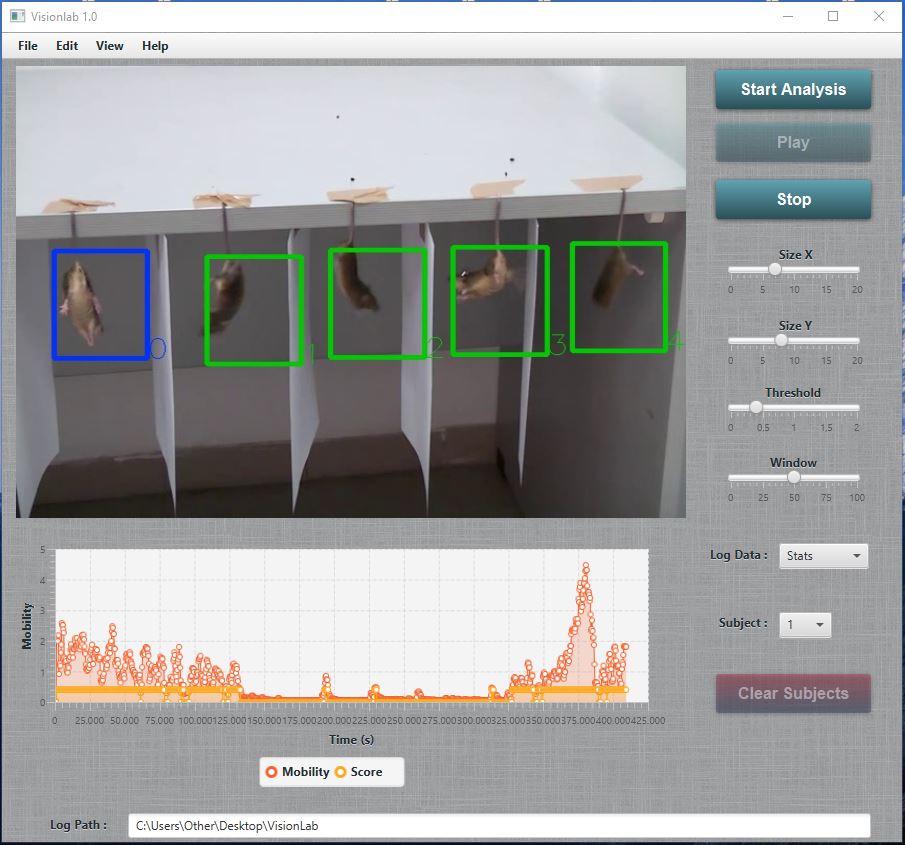
Εικόνα 12: Ένδειξη προόδου ανάλυσης

Αφού ολοκληρωθεί ο διαδικασία εμφανίζονται τα σχετικά διαγράμματα βαθμολόγησης του βίντεο τα οποία αποτελούνται από την ένταση της κίνησης (Mobility) και την ανίχνευση κίνησης του συστήματος (Score) όπως αυτά απεικονίζονται στην εικόνα 13. Ο χρήστης μπορεί να πλοηγηθεί ανάμεσα στα διαγράμματα κάθε πειραματόζωου με χρήση του επιλογέα Subject.



Εικόνα 13: Διαγράμματα αποτελεσμάτων

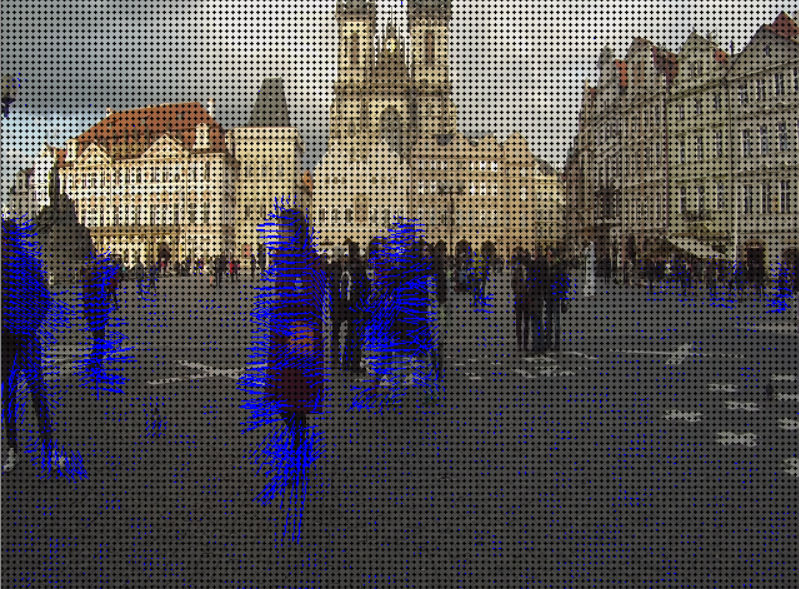
Έχοντας ολοκληρώσει την διαδικασία της ανάλυσης ο χρήστης μπορεί να επιβλέψει τα αποτελέσματα μέσω της αναπαραγωγής του βίντεο κατά την οποία η βαθμολογηση του πειράματος εμφανίζεται στον χρωματισμό του περιγράμματος κάθε πειραματόζωου (Εικόνα 14.).



Εικόνα 14:. Οπτικοποίηση σκόρ κατά την αναπαραγωγή

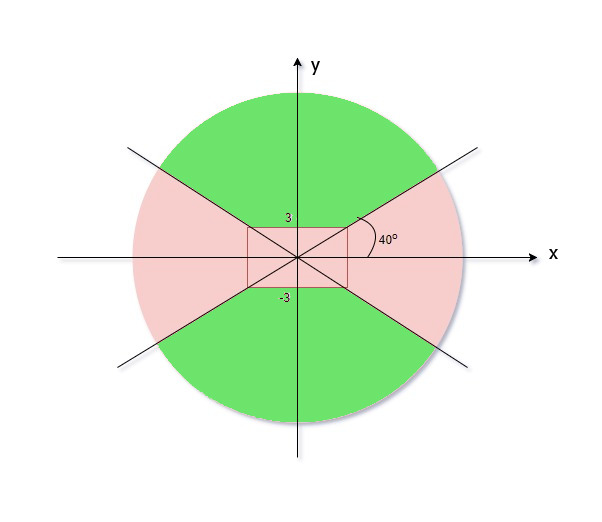
## Αλγόριθμος Ανάλυσης Πειραμάτων TST

Για την αυτόματη βαθμολόγηση πειραμάτων TST είναι απαραίτητη η αναγνώριση κίνησης του πειραματόζωου σε κάθε χρονική στιγμή, οι δυο καταστάσεις που αφορούν το πείραμα είναι η προσπάθεια διαφυγής και παύση προσπάθειας διαφυγής. Για την ανάλυση κίνησης σε μια σειρά εικόνων προσφέρεται ο αλγόριθμος ανίχνευσης οπτικής ροής Farneback ο οποίος περιλαμβάνεται στην βιβλιοθήκη OpenCV. Ο αλγόριθμος Farneback λαμβάνει ως είσοδο δυο διαδοχικά frames του βίντεο και επιστρέφει ένα διάνυσμα κίνησης για κάθε pixel της εικόνας (Εικόνα 11). Για την ανάλυση των βίντεο TST χρησιμοποιείται αυτόματη προσαρμογή μεγέθους κάθε εικόνας για να λαμβάνεται σταθερός αριθμός σημείων.



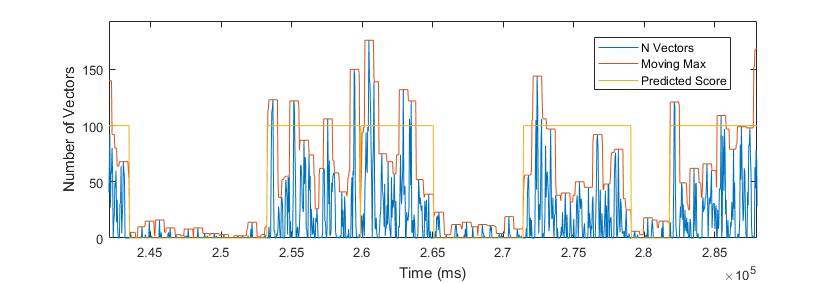
Εικόνα 11: Ανίχνευση οπτικής ροής μέσω του αλγορίθμου Farneback

Για τα δεδομένα των πειραμάτων TST η εξαγωγή των διανυσμάτων κίνησης δεν αρκεί για τον προσδιορισμό της προσπάθειας διαφυγής καθώς λόγω της ανάρτησης του πειραματόζωου από την ουρά εμφανίζονται ταλαντώσεις μετά την ακινητοποίηση του. Το πρόβλημα αυτό μπορεί να λυθεί φιλτράροντας τα διανύσματα κίνησης ανάλογα με την διεύθυνση και την ένταση τους όπως φαίνεται στην εικόνα 12.



Εικόνα 12: Φιλτράρισμα διανυσμάτων (πράσινο: αποδεκτά, κόκκινο: απορριπτέα)

Έχοντας φιλτράρει τα διανύσματα κίνησης μπορεί να χρησιμοποιηθεί ο αριθμός αποδεκτών διανυσμάτων ως μέτρο της προσπάθειας διαφυγής του πειραματόζωου. Για την απόρριψη των διακυμάνσεων που παρατηρούνται στην κίνηση υπολογίζεται η μέγιστη τιμή εντός ενός κυλιόμενου παραθύρου το μέγεθος του οποίου μπορεί να ρυθμιστεί από τον χρήστη. Τέλος ορίζεται ένα ρυθμιζόμενο κατώφλι ανίχνευσης κίνησης το ποιο καθορίζει την τελική βαθμολόγηση κίνησης (Εικόνα 13).



Εικόνα 13: Δεδομένα κίνησης

# Πειραματικά Αποτελέσματα

## Πειραματικά Δεδομένα

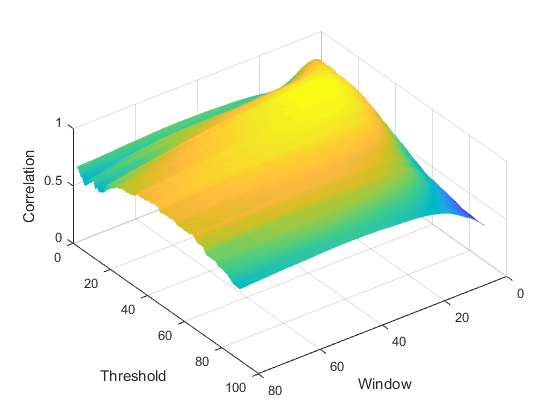
Για την ανάπτυξη του συστήματος δίνεται από το εργαστήριο Φαρμακολογίας της Ιατρικής Σχολής Αθηνών μια σειρά καταγραφών βαθμολογημένων πειραμάτων TST. Η σειρά αυτή περιλαμβάνει 6 πειράματα σε διαφορετικά τρωκτικά τα οποία καταγράφονται σε βίντεο ανάλυσης 640x360, ρυθμού 24fps και διάρκειας 6 λεπτών. Τα βίντεο συνοδεύονται από τα αποτελέσματα δυο βαθμολογητών στην μορφή χρονοσειρών και συνολικού χρόνου κίνησης.



Εικόνα 16: Βίντεο πειραμάτων TST

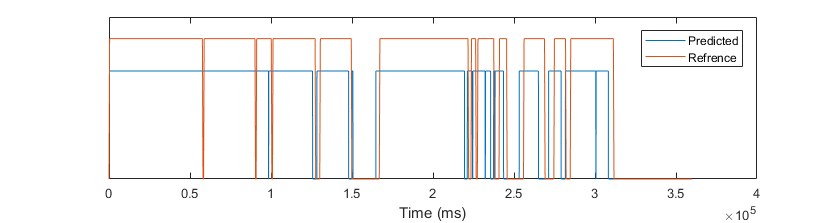
## Αποτελέσματα

Για τις δυο παραμέτρους του αλγορίθμου (μέγεθος παραθύρου, κατώφλι κίνησης), επιλέγονται οι τιμές τους εξετάζοντας έναν μεγάλο αριθμό συνδυασμών και υπολογίζοντας την συσχέτιση με τους πραγματικούς βαθμολογητές στον συνολικό χρόνο κίνησης. Όπως φαίνεται στην εικόνα 14 η βέλτιστη συσχέτιση επιτυγχάνεται για μέγεθος παραθύρου 16 και για κατώφλι 23.



Εικόνα 14: Απόδοση αλγορίθμου συναρτήσει των παραμέτρων

Στην εικόνα 15 απεικονίζεται η προβλεπόμενη βαθμολόγηση του πειράματος σε σύγκριση με την πραγματική και παρατηρείται ότι το σύστημα επιτυγχάνει να ανιχνεύει σωστά την διάρκεια κάθε κίνησης εμφανίζοντας μια μικρή προήγηση στον χρόνο που πιθανώς οφείλεται στον χρόνο αντίδρασης του βαθμολογητή.



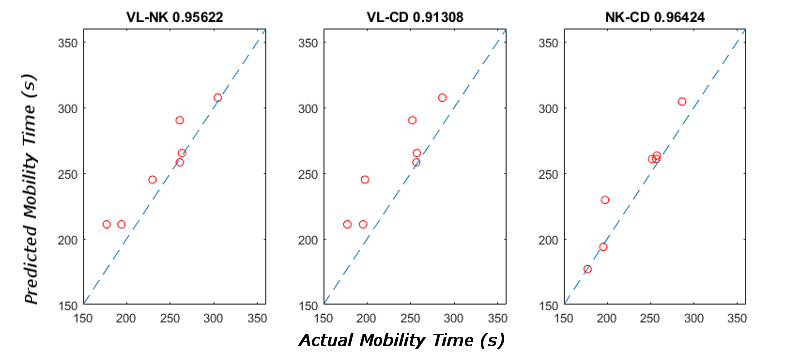
Εικόνα 15: Σύγκριση πραγματικού (πορτοκαλί) με προβλεπόμενο (μπλέ) σκορ

Για την εκτίμηση της απόδοσης του τελικού συστήματος εξετάζεται η συσχέτιση των εκτιμήσεων του συνολικού χρόνου κίνησης του κάθε πειραματόζωου που υπολογίζει το αυτοματοποιημένο σύστημα (VL) με αυτόν που προκύπτει από τους δυο βαθμολογητές (NK, CD). Στον πίνακα 1 φαίνονται οι συντελεστές συσχέτισης ανά δυο βαθμολογητές και φαίνεται ότι η συσχέτιση του συστήματος με κάθε βαθμολογητή πλησιάζει την συσχέτιση μεταξύ των δυο βαθμολογητών αναφοράς (ΝΚ,CD).

|  |  |  |  |
| --- | --- | --- | --- |
|  | Αυτόματο Σύστημα (VL) | Βαθμολογητής 1 (NK) | Βαθμολογητής 2 (CD) |
| Αυτόματο Σύστημα (VL) | 1 | 0.95622 | 0.91308 |
| Βαθμολογητής 1 (NK) | 0.95622 | 1 | 0.96424 |
| Βαθμολογητής 2 (CD) | 0.91308 | 0.96424 | 1 |

Πίνακας 1: Συντελεστές συσχέτισης μεταξύ κάθε ζεύγους βαθμολογητών (γραμμή, στήλη)

Στην εικόνα 17 φαίνονται οι προβλεπόμενες τιμές συναρτήσει των πραγματικών μεταξύ του συστήματος και κάθε βαθμολογητή και μεταξύ των δυο βαθμολογητών. Από το σχήμα φαίνεται ότι το αυτόματο σύστημα παρουσιάζει σφάλματα μερικών δευτερολέπτων τα οποία όμως δεν υπερβαίνουν σε διάρκεια τα σφάλματα που εμφανίζονται μεταξύ βαθμολογητών.



Εικόνα 17: Διάγραμμα αποτελεσμάτων αυτοματοποιημένου συστήματος (VL) σε σύγκριση με των βαθμολογητών (NK,CD)

# Συμπεράσματα – Βελτιώσεις

## Συμπεράσματα

Από την χρήση του τελικού λογισμικού φαίνεται ότι το σύστημα προσφέρει ένα φιλικό προς τον χρήστη περιβάλλον με κατανοητή ροή εργασίας και απλό σύστημα διαχείρισης αρχείων. Το λογισμικό επίσης δεν απαιτεί εγκατάσταση με αποτέλεσμα να μην απαιτεί συγκεκριμένο σταθμό εργασίας και να είναι προσιτό σε οποιονδήποτε χρήστη. Επίσης η χρονική επίδοση του συστήματος εξαρτάται από τους διαθέσιμους υπολογιστικούς πόρους με αποτέλεσμα η ανάλυση να μπορεί να ολοκληρωθεί σε μερικά δευτερόλεπτα σε σύγχρονους επεξεργαστές.

Τα αποτελέσματα βαθμολόγησης του πειράματος προσεγγίζουν αυτά των πραγματικών βαθμολογητών και εμφανίζουν επαναληψιμότητα σε κάθε εκτέλεση. Σε περίπτωση που ο χρήστης επιθυμεί την προσαρμογή του τρόπου βαθμολόγησης προσφέρεται δυνατότητα μεταβολής των παραμέτρων του αλγορίθμου μέσω του γραφικού περιβάλλοντος. Επίσης για την άμεση αξιολόγηση των αποτελεσμάτων του αλγορίθμου, μετά από κάθε ανάλυση ενός βίντεο εμφανίζονται διαγράμματα της κινητικότητας του κάθε πειραματόζωου και των αποφάσεων του συστήματος. Επιπλέον ως οπτικός τρόπος αξιολόγησης των αποτελεσμάτων δίνεται η δυνατότητα επικάλυψης των αποφάσεων βαθμολόγησης στο αρχικό βίντεο.

## Βελτιώσεις

Ο αλγόριθμος βαθμολόγησης μπορεί να βελτιωθεί χρησιμοποιώντας προηγμένες μεθόδους ταξινόμησης καταστάσεων όπως τα αναδρομικά νευρωνικά δίκτυα (RNNs). Η εκπαίδευση ενός τέτοιου ταξινομητή θα απαιτούσε μεγάλο αριθμό βαθμολογημένων πειραμάτων. Για την την αναγνώριση της κατάστασης του πειραματόζωου θα μπορούσαν να χρησιμοποιηθούν επιπλέον φυσικά δεδομένα όπως φωνητικές εκπομπές και δονήσεις στην βάση ανάρτησης τα οποία μπορούν να καταγραφούν από αισθητήρες

Το σύστημα λόγω της οργανωμένης δομής του επιτρέπει την επέκταση της εφαρμογής για την υποστήριξη επιπλέον συμπεριφοριφόρικών πειραμάτων στο ίδιο γραφικό περιβάλλον χρήστη. Επιπλέον το γραφικό περιβάλλον θα μπορούσε να ενσωματώσει δυνατότητα χειροκίνητης βαθμολόγησης βίντεο για την διευκόλυνση των χρηστών που δεν επιθυμούν την χρήση αυτοματοποιημένης μεθόδου.

Το επόμενο βήμα στην επέκταση του συστήματος θα ήταν η δημιουργία δικτυακής πλατφόρμας βαθμολόγησης και αποθήκευσης πειραματικών βίντεο η οποία θα αποτελούσε ένα χρήσιμο εργαλείο για την παρακολούθηση και τον διαμοιρασμό φαρμακολογικών μελετών.

# Βιβλιογραφία

[1] Can, A., Dao, D., Terrillion, C., Piantadosi, S., Bhat, S. and Gould, T. (2011). The Tail Suspension Test. Journal of Visualized Experiments, (58).

[2] Juszczak, G., Sliwa, A., Wolak, P., Tymosiakzueliniska, A., Lisowski, P. and Swiergiel, A. (2006). The usage of video analysis system for detection of immobility in the tail suspension test in mice. Pharmacology Biochemistry and Behavior, 85(2), pp.332-338.

[3] Lad, H., Liu, L., Payá-Cano, J., Fernandes, C. and Schalkwyk, L. (2007). Quantitative traits for the tail suspension test: automation, optimization, and BXD RI mapping. Mammalian Genome, 18(6-7), pp.482-491.

[4] JetBrains. (2017). IntelliJ IDEA: The Java IDE for Professional Developers by JetBrains. [online] Available at: https://www.jetbrains.com/idea/ [Accessed 15 Oct. 2017].

[5] Docs.oracle.com. (2017). JavaFX 2 Tutorials and Documentation. [online] Available at: http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm [Accessed 15 Oct. 2017].

[6] Opencv.org. (2017). OpenCV library. [online] Available at: https://opencv.org/ [Accessed 15 Oct. 2017].

[7] Commons.apache.org. (2017). Commons Math: The Apache Commons Mathematics Library. [online] Available at: http://commons.apache.org/proper/commons-math/ [Accessed 15 Oct. 2017]. (Commons.apache.org, 2017)

# Κώδικας συστήματος

## Main.java

package gr**.**teiath**.**visionlab**;**

**import** javafx**.**application**.**Application**;**

**import** javafx**.**embed**.**swing**.**SwingFXUtils**;**

**import** javafx**.**event**.**ActionEvent**;**

**import** javafx**.**fxml**.**FXML**;**

**import** javafx**.**fxml**.**FXMLLoader**;**

**import** javafx**.**fxml**.**Initializable**;**

**import** javafx**.**scene**.**Parent**;**

**import** javafx**.**scene**.**Scene**;**

**import** javafx**.**scene**.**chart**.**AreaChart**;**

**import** javafx**.**scene**.**control**.\*;**

**import** javafx**.**scene**.**image**.**Image**;**

**import** javafx**.**scene**.**image**.**ImageView**;**

**import** javafx**.**scene**.**input**.**DragEvent**;**

**import** javafx**.**scene**.**input**.**Dragboard**;**

**import** javafx**.**scene**.**input**.**MouseEvent**;**

**import** javafx**.**scene**.**input**.**TransferMode**;**

**import** javafx**.**stage**.**FileChooser**;**

**import** javafx**.**stage**.**Stage**;**

**import** org**.**opencv**.**core**.\*;**

**import** org**.**opencv**.**imgcodecs**.**Imgcodecs**;**

**import** org**.**opencv**.**imgproc**.**Imgproc**;**

**import** org**.**opencv**.**videoio**.**VideoCapture**;**

**import** javax**.**imageio**.**ImageIO**;**

**import** java**.**awt**.**image**.**BufferedImage**;**

**import** java**.**io**.**ByteArrayInputStream**;**

**import** java**.**io**.**File**;**

**import** java**.**io**.**InputStream**;**

**import** java**.**net**.**URL**;**

**import** java**.**util**.\*;**

public class Main **extends** Application **implements** Initializable **{**

@FXML

Button startAnalysis**,**play**,**stop**,**clear**;**

@FXML

CheckMenuItem visScore**;**

@FXML

ImageView view**;**

@FXML

Label labelDrag**,**labelClick**;**

@FXML

ProgressIndicator progress**;**

@FXML

AreaChart**<**Number**,**Number**>** chart**;**

@FXML

Slider slidThres**,**slidWind**,**slidYsize**,**slidXsize**;**

@FXML

ComboBox choiceLog**,**choiceSub**;**

@FXML

TextField logPath**;**

Stage stage**;**

public static volatile VideoManager mVideoManager**;**

public static volatile VideoAnalyzer mVideoAnalyzer**;**

public long startTime**;**

public String filePath**,** filename**;**

public Mat previewframe**=new** Mat**();**

public boolean areaSelected**=false,**areasFinalized**=false,**fileLoaded**=false,**analysisDataAvailable**=false;**

public ArrayList**<**Rect**>** measAreas **=** **new** ArrayList**<**Rect**>();**

public ArrayList**<**AreaChart**.**Series**>** analysisDataM **=** **new** ArrayList**<**AreaChart**.**Series**>();**

public ArrayList**<**AreaChart**.**Series**>** analysisDataS **=** **new** ArrayList**<**AreaChart**.**Series**>();**

public ArrayList**<**double**[]>** scoreData**;**

@Override

public void start**(**Stage primaryStage**)** **throws** Exception **{**

Parent root **=** FXMLLoader**.**load**(**getClass**().**getResource**(**"layout.fxml"**));**

primaryStage**.**setTitle**(**"VisionLab 1.0"**);**

Scene scene **=** **new** Scene**(**root**,** 900**,**710**);**

primaryStage**.**setScene**(**scene**);**

scene**.**getStylesheets**().**add**(**Main**.**class**.**getResource**(**"styles.css"**).**toExternalForm**());**

primaryStage**.**show**();**

stage **=** primaryStage**;**

**}**

@Override

public void stop**()** **{**

**if** **(**mVideoManager**!=null){**

mVideoManager**.**stop**();**

**}**

**}**

@FXML

public void initialize**(**URL location**,** ResourceBundle resources**){**

logPath**.**setText**(**System**.**getProperty**(**"user.home"**)+** File**.**separator**+**"Desktop"**+**File**.**separator**+**"VisionLab"**);**

**}**

public void onStartAnalysis**(**ActionEvent actionEvent**)** **{**

onStop**(new** ActionEvent**());**

areasFinalized **=** **true;**

startAnalysis**.**setDisable**(true);**

labelClick**.**setVisible**(false);**

play**.**setDisable**(true);**

progress**.**setVisible**(true);**

progress**.**setProgress**(-**1**);**

mVideoAnalyzer **=** **new** VideoAnalyzer**(**filePath**);**

mVideoAnalyzer**.**start**(this,**filename**);**

view**.**setImage**(null);**

view**.**setVisible**(false);**

clear**.**setDisable**(true);**

**}**

public void onPlay**(**ActionEvent actionEvent**)** **{**

areasFinalized **=** **true;**

play**.**setDisable**(true);**

mVideoManager **=** **new** VideoManager**(**filePath**);**

mVideoManager**.**start**(this);**

view**.**setVisible**(true);**

stop**.**setDisable**(false);**

startTime **=** System**.**currentTimeMillis**();**

labelClick**.**setVisible**(false);**

progress**.**setVisible**(false);**

clear**.**setDisable**(true);**

**}**

public void onStop**(**ActionEvent actionEvent**)** **{**

stop**.**setDisable**(true);**

labelClick**.**setVisible**(false);**

**if(**mVideoManager**!=null)**mVideoManager**.**stop**();**

mVideoManager**=null;**

**if(**mVideoAnalyzer**!=null)**mVideoAnalyzer**.**stop**();**

mVideoAnalyzer**=null;**

view**.**setImage**(null);**

play**.**setDisable**(false);**

startAnalysis**.**setDisable**(false);**

loadpreview**();**

view**.**setVisible**(true);**

clear**.**setDisable**(false);**

System**.**gc**();**

**}**

public void onMouseClick**(**MouseEvent mouseEvent**){**

**if(**fileLoaded **&&** **!**areasFinalized**)** **{**

double x **=** **(**mouseEvent**.**getX**()** **/** view**.**getFitWidth**());**

double y **=** mouseEvent**.**getY**()** **/** view**.**getFitHeight**();**

double ratio **=** view**.**getFitWidth**()** **/** view**.**getFitHeight**();**

Point startpoint**=new** Point**((**x**+**slidXsize**.**getValue**()\***0.01**)\***previewframe**.**width**(),(**y**+**slidYsize**.**getValue**()\***0.01**\***ratio**)\***previewframe**.**height**());**

Point endpoint**=new** Point**((**x**-**slidXsize**.**getValue**()\***0.01**)\***previewframe**.**width**(),(**y**-**slidYsize**.**getValue**()\***0.01**\***ratio**)\***previewframe**.**height**());**

measAreas**.**add**(new** Rect**(**startpoint**,**endpoint**));**

areaSelected**=true;**

showpreview**();**

startAnalysis**.**setDisable**(false);**

play**.**setDisable**(false);**

clear**.**setDisable**(false);**

choiceLog**.**setDisable**(false);**

choiceSub**.**setDisable**(false);**

**if(**measAreas**.**size**()>**1**)** choiceSub**.**getItems**().**add**(**Integer**.**toString**(**measAreas**.**size**()-**1**));**

**}**

**}**

public void onFileEnter**(**DragEvent event**)** **{**

**if(!**fileLoaded**)** **{**

Dragboard db **=** event**.**getDragboard**();**

**if** **(**db**.**hasFiles**())** **{**

event**.**acceptTransferModes**(**TransferMode**.**COPY**);**

**}** **else** **{**

event**.**consume**();**

**}**

**}**

**}**

public void onFileDrop**(**DragEvent event**){**

**if(!**fileLoaded**)** **{**

Dragboard db **=** event**.**getDragboard**();**

boolean success **=** **false;**

**if** **(**db**.**hasFiles**())** **{**

success **=** **true;**

**for** **(**File file **:** db**.**getFiles**())** **{**

filePath **=** file**.**getAbsolutePath**();**

**}**

openFile**();**

**}**

event**.**setDropCompleted**(**success**);**

event**.**consume**();**

**}**

**}**

public void onFileOpen**(**ActionEvent event**){**

**if(!**fileLoaded**)** **{**

FileChooser fileChooser **=** **new** FileChooser**();**

fileChooser**.**setTitle**(**"Open Experiment Video"**);**

File file **=** fileChooser**.**showOpenDialog**(**stage**);**

**if** **(**file **!=** **null)** **{**

filePath **=** file**.**getPath**();**

openFile**();**

**}**

**}**

**}**

public void openFile**(){**

labelDrag**.**setVisible**(false);**

labelClick**.**setVisible**(true);**

fileLoaded**=true;**

File f **=** **new** File**(**filePath**);**

filename**=**f**.**getName**().**substring**(**0**,**f**.**getName**().**lastIndexOf**(**'.'**));**

loadpreview**();**

**}**

public void onFileClose**(**ActionEvent actionEvent**)** **{**

**if(**fileLoaded**)** **{**

analysisDataAvailable**=false;**

analysisDataM**.**clear**();**

analysisDataS**.**clear**();**

fileLoaded **=** **false;**

areaSelected **=** **false;**

areasFinalized **=** **false;**

measAreas**.**clear**();**

chart**.**getData**().**clear**();**

**if** **(**mVideoManager **!=** **null)** mVideoManager**.**stop**();**

mVideoManager**=null;**

**if(**mVideoAnalyzer**!=null)**mVideoAnalyzer**.**stop**();**

mVideoAnalyzer**=** **null;**

view**.**setImage**(null);**

view**.**setVisible**(false);**

progress**.**setVisible**(false);**

startAnalysis**.**setDisable**(true);**

stop**.**setDisable**(true);**

play**.**setDisable**(true);**

labelDrag**.**setVisible**(true);**

labelClick**.**setVisible**(false);**

System**.**gc**();**

**}**

**}**

public void onClearSubjects**(**ActionEvent actionEvent**){**

measAreas**.**clear**();**

loadpreview**();**

chart**.**getData**().**clear**();**

areaSelected**=** **false;**

areasFinalized **=** **false;**

progress**.**setVisible**(false);**

labelClick**.**setVisible**(true);**

showpreview**();**

startAnalysis**.**setDisable**(true);**

play**.**setDisable**(true);**

clear**.**setDisable**(true);**

choiceLog**.**setDisable**(true);**

choiceSub**.**setDisable**(true);**

**}**

public void onPlotDataChanged**(**ActionEvent actionEvent**)** **{**

plotData**(**choiceSub**.**getValue**().**toString**());**

**}**

public void plotData**(**String sub**)** **{**

**if** **(**analysisDataAvailable**){**

chart**.**getData**().**clear**();**

chart**.**getData**().**add**(**analysisDataM**.**get**(**Integer**.**parseInt**(**sub**)));**

chart**.**getData**().**add**(**analysisDataS**.**get**(**Integer**.**parseInt**(**sub**)));**

**}**

**}**

public void loadpreview**(){**

VideoCapture prevcap**=new** VideoCapture**(**filePath**);**

**for** **(**int i **=** 0**;** i **<** 10 **&&** prevcap**.**isOpened**();**i**++){**

prevcap**.**read**(**previewframe**);**

**}**

**if(**previewframe**.**size**().**area**()>**500000**)** Imgproc**.**pyrDown**(**previewframe**,** previewframe**);**//High res only

prevcap**.**release**();**

view**.**setVisible**(true);**

showpreview**();**

**}**

public void showpreview**(){**

**for(**int i**=**0**;** i**<**measAreas**.**size**();**i**++)** **{**

Imgproc**.**rectangle**(**previewframe**,** measAreas**.**get**(**i**).**tl**(),** measAreas**.**get**(**i**).**br**(),** **new** Scalar**(**255**,** 50**,** 0**),** 3**);**

Imgproc**.**putText**(**previewframe**,**Integer**.**toString**(**i**),**measAreas**.**get**(**i**).**br**(),**Core**.**FONT\_HERSHEY\_SIMPLEX**,**1**,new** Scalar**(**255**,** 50**,** 0**));**

**}**

MatOfByte bytemat **=** **new** MatOfByte**();**

Imgcodecs**.**imencode**(**".jpg"**,** previewframe**,** bytemat**);**

byte**[]** bytes **=** bytemat**.**toArray**();**

InputStream in **=** **new** ByteArrayInputStream**(**bytes**);**

Image prev**;**

**try** **{**

BufferedImage img **=** ImageIO**.**read**(**in**);**

prev **=** SwingFXUtils**.**toFXImage**(**img**,** **null);**

onImage**(**prev**);**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public void onImage**(**Image image**){**

view**.**setImage**(**image**);**

**}**

public static void main**(**String**[]** args**)** **{**

System**.**loadLibrary**(**"opencv\_java300"**);**

System**.**loadLibrary**(**"opencv\_ffmpeg300\_64"**);**

launch**(**args**);**

**}**

**}**

## VideoAnalyzer.java

package gr**.**teiath**.**visionlab**;**

**import** javafx**.**application**.**Platform**;**

**import** javafx**.**event**.**ActionEvent**;**

**import** javafx**.**scene**.**chart**.**AreaChart**;**

**import** org**.**apache**.**commons**.**math3**.**stat**.**StatUtils**;**

**import** org**.**apache**.**commons**.**math3**.**stat**.**descriptive**.**DescriptiveStatistics**;**

**import** org**.**opencv**.**core**.\*;**

**import** org**.**opencv**.**imgproc**.**Imgproc**;**

**import** org**.**opencv**.**video**.**Video**;**

**import** org**.**opencv**.**videoio**.**VideoCapture**;**

**import** org**.**opencv**.**videoio**.**Videoio**;**

**import** java**.**io**.**BufferedWriter**;**

**import** java**.**io**.**File**;**

**import** java**.**io**.**FileWriter**;**

**import** java**.**text**.**SimpleDateFormat**;**

**import** java**.**util**.**ArrayList**;**

**import** java**.**util**.**Date**;**

public class VideoAnalyzer **{**

public volatile VideoCapture Capture**;**

double curprogress**;**

boolean endvideo**=false;**

double framerate**,**framecount**,**duration**;**

Main UI**;**

String filename**;**

BufferedWriter logger**=null;**

Thread CameraManager**=new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Thread**.**currentThread**().**setPriority**(**Thread**.**MAX\_PRIORITY**);**

//Initialization

double**[]** time **=** **new** double**[(**int**)**framecount**];**

ArrayList**<**double**[]>** vecPower **=** **new** ArrayList**<**double**[]>();**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** vecPower**.**add**(**i**,new** double**[(**int**)**framecount**]);**

ArrayList**<**double**[]>** vecNum **=** **new** ArrayList**<**double**[]>();**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** vecNum**.**add**(**i**,new** double**[(**int**)**framecount**]);**

Mat frame **=** **new** Mat**();**

Mat prevframe **=** Mat**.**zeros**((**int**)**Capture**.**get**(**Videoio**.**CAP\_PROP\_FRAME\_HEIGHT**),** **(**int**)**Capture**.**get**(**Videoio**.**CAP\_PROP\_FRAME\_WIDTH**),**CvType**.**CV\_8UC3**);**

**while** **(**Capture**.**isOpened**()&&!**endvideo**)** **{**

Mat image **=** **new** Mat**();**

**if(!**Capture**.**isOpened**())break;**

Capture**.**read**(**image**);**

**if(!**image**.**empty**())** **{**

image**.**copyTo**(**frame**);**

int idx **=** **(**int**)**Capture**.**get**(**Videoio**.**CAP\_PROP\_POS\_FRAMES**)-**1**;**

time**[**idx**]** **=** Capture**.**get**(**Videoio**.**CAP\_PROP\_POS\_MSEC**);**

//Processing

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** **{**

Mat measurementArea**=**frame**.**submat**(**UI**.**measAreas**.**get**(**i**));**

Mat prevMeasurementArea**=**prevframe**.**submat**(**UI**.**measAreas**.**get**(**i**));**

Imgproc**.**cvtColor**(**measurementArea**,**measurementArea**,**Imgproc**.**COLOR\_BGR2GRAY**);**

Imgproc**.**cvtColor**(**prevMeasurementArea**,** prevMeasurementArea**,** Imgproc**.**COLOR\_BGR2GRAY**);**

MatOfPoint flow **=** **new** MatOfPoint**();**

Mat graysmall **=** **new** Mat**();**

Mat prevsmall **=** **new** Mat**();**

Imgproc**.**resize**(**measurementArea**,**graysmall**,new** Size**(**300**,**300**));**

Imgproc**.**resize**(**prevMeasurementArea**,**prevsmall**,new** Size**(**300**,**300**));**

Double motion **=** 0.0**;**

Integer points **=** 0**;**

Video**.**calcOpticalFlowFarneback**(**prevsmall**,** graysmall**,** flow**,** 0.5**,** 3**,** 15**,** 3**,** 5**,** 1.2**,** 0**);**

**for** **(**int y **=** 0**;** y **<** flow**.**rows**();** y **+=** 6**){**

**for** **(**int x **=** 0**;** x **<** flow**.**cols**();** x **+=** 6**)** **{**

double**[]** data**=** flow**.**get**(**y**,**x**);**

Point pt **=** **new** Point**(**data**[**0**],**data**[**1**]);**

**if(** **(**Math**.**abs**(((**double**)** pt**.**y**) /((**double**)**pt**.**x**)))** **>** Math**.**tan**(**40**\***Math**.**PI**/**180**)** **&&** Math**.**abs**(**pt**.**y**)** **>** 3**)** **{**

motion**+=**Math**.**sqrt**(**Math**.**pow**(**pt**.**x**,**2**)+**Math**.**pow**(**pt**.**x**,**2**));**

points**++;**

**}**

**}**

**}**

vecPower**.**get**(**i**)[**idx**]** **=** motion**;**

vecNum**.**get**(**i**)[**idx**]** **=** points**;**

**}**

image**.**copyTo**(**prevframe**);**

setProgress**(**time**[**idx**]/(**duration**\***1000**));**

System**.**gc**();**

**}**

**else** **break;**

**}**

setProgress**(-**1**);**

//Process Output Data

int window **=** **(**int**)** UI**.**slidWind**.**getValue**();**

double threshold **=** UI**.**slidThres**.**getValue**();**

int n **=** **(**int**)**framecount**;**

ArrayList**<**double**[]>** movMax **=** **new** ArrayList**<**double**[]>();**

ArrayList**<**double**[]>** behavData **=** **new** ArrayList**<**double**[]>();**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++){**

movMax**.**add**(**i**,new** double**[(**int**)**framecount**]);**

behavData**.**add**(**i**,new** double**[(**int**)**framecount**]);**

**}**

ArrayList**<**Double**>** immobilityData **=** **new** ArrayList**<**Double**>();**

**for(**int id**=**0**;** id**<**UI**.**measAreas**.**size**();**id**++)** **{**

DescriptiveStatistics stats **=** **new** DescriptiveStatistics**();**

stats**.**setWindowSize**(**40**);**

**for(**int i **=** 0**;** i**<**n**;** i**++** **){**

stats**.**addValue**(**vecNum**.**get**(**id**)[**i**]);**

movMax**.**get**(**id**)[**i**]** **=** stats**.**getMax**();**

**}**

**for(**int i **=** 0**;** i**<**n**;** i**++** **)** **if** **(**movMax**.**get**(**id**)[**i**]** **>=** threshold**)** behavData**.**get**(**id**)[**i**]** **=** 1**;**

int zeros**=**0**;**

**for(**int i **=** 0**;** i**<**n**;** i**++** **)** **if** **(**behavData**.**get**(**id**)[**i**]** **==** 0**)** zeros**++;**

immobilityData**.**add**(**id**,**zeros**/**framerate**);**

**}**

//Show Output Data

UI**.**analysisDataM**.**clear**();**

UI**.**analysisDataS**.**clear**();**

**for(**int id**=**0**;** id**<**UI**.**measAreas**.**size**();**id**++)** **{**

UI**.**analysisDataM**.**add**(**id**,new** AreaChart**.**Series**<**Number**,**Number**>());**

UI**.**analysisDataM**.**get**(**id**).**setName**(**"Mobility"**);**

UI**.**analysisDataM**.**get**(**id**).**getData**().**add**(new** AreaChart**.**Data**<**Number**,** Number**>(**time**[**0**],** 0**));**

**for(**int i **=** 1**;** i**<**n**;** i**+=**10 **)** UI**.**analysisDataM**.**get**(**id**).**getData**().**add**(new** AreaChart**.**Data**<**Number**,** Number**>(**time**[**i**],** vecNum**.**get**(**id**)[**i**]));**

UI**.**analysisDataS**.**add**(**id**,new** AreaChart**.**Series**<**Number**,**Number**>());**

UI**.**analysisDataS**.**get**(**id**).**setName**(**"Score"**);**

**for(**int i **=** 0**;** i**<**n**;** i**+=**10 **)** UI**.**analysisDataS**.**get**(**id**).**getData**().**add**(new** AreaChart**.**Data**<**Number**,** Number**>(**time**[**i**],** threshold**\***behavData**.**get**(**id**)[**i**]));**

**}**

UI**.**analysisDataAvailable**=true;**

**try{**

Thread**.**sleep**(**100**);**

**}catch** **(**InterruptedException e**){}**

Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

UI**.**onPlotDataChanged**(new** ActionEvent**());**

**}**

**});**

//Log Output Data

SimpleDateFormat sdf **=** **new** SimpleDateFormat**(**"dd-MM-yyyy\_HH-mm-ss"**);**

String logfilename **=** filename **+** "\_" **+** sdf**.**format**(new** Date**())** **+** ".txt"**;**

String path **=** UI**.**logPath**.**getText**();**

File f **=** **new** File**(**path**);**

f**.**mkdirs**();**

**try{**

logger **=** **new** BufferedWriter**(new** FileWriter**(new** File**(**path**,**logfilename**)));**

**}**

**catch(**Exception e**){**e**.**printStackTrace**();}**

**if(**UI**.**choiceLog**.**getValue**().**equals**(**"Stats"**)){**

**try{**

logger**.**append**(**"Total Immobility (s)"**);**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** **{**

logger**.**append**(**String**.**format**(**"\t%.2f"**,**immobilityData**.**get**(**i**)));**

**}**

logger**.**newLine**();**

logger**.**append**(**"Total Mobility (s)"**);**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** **{**

logger**.**append**(**String**.**format**(**"\t%.2f"**,**duration**-**immobilityData**.**get**(**i**)));**

**}**

logger**.**newLine**();**

logger**.**append**(**String**.**format**(**"Area Width\t%.2f"**,**UI**.**slidXsize**.**getValue**()));**

logger**.**newLine**();**

logger**.**append**(**String**.**format**(**"Area Height\t%.2f"**,**UI**.**slidYsize**.**getValue**()));**

logger**.**newLine**();**

logger**.**append**(**String**.**format**(**"Window\t%d"**,**window**));**

logger**.**newLine**();**

logger**.**append**(**String**.**format**(**"Threshold\t%.1f"**,**threshold**));**

logger**.**newLine**();**

**}**

**catch** **(**Exception e**){}**

**}**

**else** **if(**UI**.**choiceLog**.**getValue**().**equals**(**"Timeline"**)){**

**for(**int ind**=**0 **;** ind**<**time**.**length **;** ind**++){**

**try{**

logger**.**append**(**String**.**format**(**"%.2f"**,**time**[**ind**]));**

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** **{**

logger**.**append**(**String**.**format**(**"\t%.0f"**,**behavData**.**get**(**i**)[**ind**]));**

logger**.**append**(**String**.**format**(**"\t%.0f"**,**vecNum**.**get**(**i**)[**ind**]));**

logger**.**append**(**String**.**format**(**"\t%.2f"**,**vecPower**.**get**(**i**)[**ind**]));**

**}**

logger**.**newLine**();**

**}**

**catch** **(**Exception e**){}**

**}**

**}**

**try{**

logger**.**close**();**

**}**

**catch(**Exception e**){}**

UI**.**scoreData**=**behavData**;**

Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

UI**.**clear**.**setDisable**(false);**

**}**

**});**

setProgress**(**1**);**

System**.**gc**();**

**if(!**endvideo**)** Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

UI**.**onStop**(new** ActionEvent**());**

**}**

**});**

**}**

public void setProgress**(**double prog**){**

curprogress**=**prog**;**

Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

**try** **{**

UI**.**progress**.**setProgress**(**curprogress**);**

**}** **catch** **(**Exception e**)** **{**

**}**

**}**

**});**

**}**

**});**

public VideoAnalyzer**(**String file**){**

Capture**=** **new** VideoCapture**(**file**);**

framerate **=** Capture**.**get**(**Videoio**.**CAP\_PROP\_FPS**);**

framecount **=** Capture**.**get**(**Videoio**.**CAP\_PROP\_FRAME\_COUNT**);**

duration **=** **(**framecount**/**framerate**);**

//System.out.println("Framerate: " + framerate);

//System.out.println("Frames: " + framecount);

//System.out.println("Duration: " + duration+"ms");

**}**

public VideoAnalyzer**(**Integer cam**){**

Capture**=** **new** VideoCapture**(**cam**);**

framerate**=**24**;**

**}**

public void start**(**Main main**,**String filename**)** **{**

**this.**filename**=**filename**;**

CameraManager**.**start**();**

UI**=**main**;**

**}**

public void stop**()** **{**

endvideo**=true;**

**try** **{**Thread**.**sleep**(**50**);}**

**catch** **(**InterruptedException e**){**e**.**printStackTrace**();}**

UI**=null;**

Capture**.**release**();**

**try{**

logger**.**close**();**

**}**

**catch(**Exception e**){}**

**}**

**}**

## VideoManager.java

package gr**.**teiath**.**visionlab**;**

**import** javafx**.**application**.**Platform**;**

**import** javafx**.**embed**.**swing**.**SwingFXUtils**;**

**import** javafx**.**event**.**ActionEvent**;**

**import** javafx**.**scene**.**image**.**Image**;**

**import** org**.**opencv**.**core**.\*;**

**import** org**.**opencv**.**imgcodecs**.**Imgcodecs**;**

**import** org**.**opencv**.**imgproc**.**Imgproc**;**

**import** org**.**opencv**.**videoio**.**VideoCapture**;**

**import** org**.**opencv**.**videoio**.**Videoio**;**

**import** javax**.**imageio**.**ImageIO**;**

**import** java**.**awt**.**image**.**BufferedImage**;**

**import** java**.**io**.**ByteArrayInputStream**;**

**import** java**.**io**.**InputStream**;**

public class VideoManager **{**

public volatile VideoCapture Capture**;**

Image image**;**

boolean endvideo**=false;**

double framerate**,**framecount**,**duration**;**

Main UI**;**

Thread CameraManager**=new** Thread**(new** Runnable**()** **{**

@Override

public void run**()** **{**

Thread**.**currentThread**().**setPriority**(**Thread**.**MAX\_PRIORITY**);**

//Initialization

Mat frame**=new** Mat**();**

int framepos**=**0**;**

**while** **(**Capture**.**isOpened**()&&!**endvideo**)** **{**

Mat image **=** **new** Mat**();**

**if(!**Capture**.**isOpened**())break;**

Capture**.**read**(**image**);**

**if(!**image**.**empty**()&&** UI**!=null)** **{**

**if(**image**.**size**().**area**()>**500000**)** Imgproc**.**pyrDown**(**image**,**image**);**//High res only

long startproc**=**System**.**currentTimeMillis**();**

image**.**copyTo**(**frame**);**

//Processing

**for(**int i**=**0**;** i**<**UI**.**measAreas**.**size**();**i**++)** **{**

Imgproc**.**rectangle**(**frame**,** UI**.**measAreas**.**get**(**i**).**tl**(),** UI**.**measAreas**.**get**(**i**).**br**(),** **new** Scalar**(**255**,** 50**,** 0**),** 3**);**

Imgproc**.**putText**(**frame**,**Integer**.**toString**(**i**),**UI**.**measAreas**.**get**(**i**).**br**(),**Core**.**FONT\_HERSHEY\_SIMPLEX**,**1**,new** Scalar**(**255**,** 50**,** 0**));**

**if(**UI**.**analysisDataAvailable**&&**UI**.**visScore**.**isSelected**()){**

**if(**UI**.**scoreData**.**get**(**i**)[**framepos**]>**0**){**

Imgproc**.**rectangle**(**frame**,** UI**.**measAreas**.**get**(**i**).**tl**(),** UI**.**measAreas**.**get**(**i**).**br**(),** **new** Scalar**(**0**,** 200**,** 0**),** 3**);**

Imgproc**.**putText**(**frame**,**Integer**.**toString**(**i**),**UI**.**measAreas**.**get**(**i**).**br**(),**Core**.**FONT\_HERSHEY\_SIMPLEX**,**1**,new** Scalar**(**0**,** 200**,** 0**));**

**}**

**}**

**}**

framepos**++;**

//Show an image

show**(**frame**);**

long proctime**=**System**.**currentTimeMillis**()-**startproc**;**

long delay**=**Math**.**round**(**1000 **/** framerate**)-**proctime**;**

**if(**delay**<**0**)**delay**=**0**;**

cvWait**(**delay**);**

//System.gc();

**}**

**else** **break;**

**}**

System**.**gc**();**

Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

UI**.**clear**.**setDisable**(false);**

**}**

**});**

**if(!**endvideo**)** Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

UI**.**onStop**(new** ActionEvent**());**

**}**

**});**

**}**

public void cvWait**(**long time**){**

**try** **{**

Thread**.**sleep**(**time**);**

**}** **catch** **(**InterruptedException e**)** **{**

e**.**printStackTrace**();**

**}**

**}**

public void show**(**Mat frame**){**

Mat showframe**=new** Mat**();**

frame**.**copyTo**(**showframe**);**

MatOfByte bytemat **=** **new** MatOfByte**();**

Imgcodecs**.**imencode**(**".jpg"**,** showframe**,** bytemat**);**

byte**[]** bytes **=** bytemat**.**toArray**();**

InputStream in **=** **new** ByteArrayInputStream**(**bytes**);**

**try** **{**

BufferedImage img **=** ImageIO**.**read**(**in**);**

image **=** SwingFXUtils**.**toFXImage**(**img**,** **null);**

**}** **catch** **(**Exception e**)** **{**

e**.**printStackTrace**();**

**}**

Platform**.**runLater**(new** Runnable**()** **{**

@Override

public void run**()** **{**

**try{**UI**.**onImage**(**image**);}**

**catch** **(**NullPointerException e**){}**

**}**

**});**

**}**

**});**

public VideoManager**(**String file**){**

Capture**=** **new** VideoCapture**(**file**);**

framerate **=** Capture**.**get**(**Videoio**.**CAP\_PROP\_FPS**);**

framecount **=** Capture**.**get**(**Videoio**.**CAP\_PROP\_FRAME\_COUNT**);**

duration **=** framecount**/**framerate**;**

//System.out.println("Framerate: " + framerate);

//System.out.println("Frames: " + framecount);

//System.out.println("Duration: " + duration/60);

**}**

public VideoManager**(**Integer cam**){**

Capture**=** **new** VideoCapture**(**cam**);**

framerate**=**24**;**

**}**

public void start**(**Main main**)** **{**

CameraManager**.**start**();**

UI**=**main**;**

**}**

public void stop**()** **{**

endvideo**=true;**

UI**=null;**

**try** **{**Thread**.**sleep**(**50**);}**

**catch** **(**InterruptedException e**){**e**.**printStackTrace**();}**

UI**=null;**

Capture**.**release**();**

**}**

**}**

## layout.fxml

<?xml version=**"1.0"** encoding=**"UTF-8"**?>

<?import java.lang.String?>

<?import java.net.URL?>

<?import javafx.collections.FXCollections?>

<?import javafx.scene.chart.AreaChart?>

<?import javafx.scene.chart.NumberAxis?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.CheckMenuItem?>

<?import javafx.scene.control.ComboBox?>

<?import javafx.scene.control.Label?>

<?import javafx.scene.control.Menu?>

<?import javafx.scene.control.MenuBar?>

<?import javafx.scene.control.MenuItem?>

<?import javafx.scene.control.ProgressIndicator?>

<?import javafx.scene.control.Slider?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.image.ImageView?>

<?import javafx.scene.layout.AnchorPane?>

<?import javafx.scene.layout.ColumnConstraints?>

<?import javafx.scene.layout.GridPane?>

<?import javafx.scene.layout.RowConstraints?>

<GridPane alignment=**"CENTER"** disable=**"false"** gridLinesVisible=**"false"** hgap=**"10.0"** prefHeight=**"710.0"** prefWidth=**"900.0"** styleClass=**"root"** vgap=**"10.0"** xmlns=**"http://javafx.com/javafx/8.0.112-ea"** xmlns:fx=**"http://javafx.com/fxml/1"** fx:controller=**"gr.teiath.visionlab.Main"**>

<children>

<AnchorPane onDragDropped=**"#onFileDrop"** onDragOver=**"#onFileEnter"** pickOnBounds=**"false"** prefHeight=**"710.0"** prefWidth=**"900.0"** GridPane.columnIndex=**"0"** GridPane.rowIndex=**"0"**>

<children>

<ImageView id=**"imageview"** fx:id=**"view"** fitHeight=**"410.0"** fitWidth=**"670.0"** focusTraversable=**"true"** layoutX=**"14.0"** layoutY=**"34.0"** onMouseClicked=**"#onMouseClick"** pickOnBounds=**"true"** preserveRatio=**"false"** visible=**"false"** />

<ProgressIndicator fx:id=**"progress"** layoutX=**"216.0"** layoutY=**"186.0"** prefHeight=**"123.0"** prefWidth=**"282.0"** progress=**"0.0"** visible=**"false"** />

<Button id=**"start"** fx:id=**"stop"** defaultButton=**"false"** disable=**"true"** layoutX=**"714.0"** layoutY=**"148.0"** mnemonicParsing=**"false"** onAction=**"#onStop"** prefHeight=**"38.0"** prefWidth=**"155.0"** text=**"Stop"** textOverrun=**"ELLIPSIS"** />

<Button id=**"start"** fx:id=**"startAnalysis"** defaultButton=**"false"** disable=**"true"** layoutX=**"714.0"** layoutY=**"38.0"** mnemonicParsing=**"false"** onAction=**"#onStartAnalysis"** prefHeight=**"38.0"** prefWidth=**"155.0"** text=**"Start Analysis"** textOverrun=**"ELLIPSIS"** wrapText=**"false"** />

<Label id=**"labelstart"** fx:id=**"labelDrag"** layoutX=**"287.0"** layoutY=**"234.0"** prefWidth=**"140.0"** text=**"Drag file here to open"** />

<Label fx:id=**"labelClick"** layoutX=**"280.0"** layoutY=**"389.0"** prefWidth=**"155.0"** text=**"Select measurement areas"** visible=**"false"** />

<Slider fx:id=**"slidThres"** layoutX=**"722.0"** layoutY=**"368.0"** majorTickUnit=**"50.0"** max=**"200.0"** minorTickCount=**"4"** showTickLabels=**"true"** showTickMarks=**"true"** snapToTicks=**"true"** value=**"100.0"** />

<Label layoutX=**"763.0"** layoutY=**"352.0"** text=**"Threshold"** />

<Slider fx:id=**"slidWind"** layoutX=**"722.0"** layoutY=**"438.0"** majorTickUnit=**"25.0"** max=**"100.0"** minorTickCount=**"4"** showTickLabels=**"true"** showTickMarks=**"true"** snapToTicks=**"true"** value=**"40.0"** />

<Label layoutX=**"768.0"** layoutY=**"422.0"** text=**"Window"** />

<MenuBar layoutX=**"0.0"** layoutY=**"1.0"** minHeight=**"23.0"** prefHeight=**"26.0"** prefWidth=**"900.0"**>

<menus>

<Menu mnemonicParsing=**"false"** text=**"File"**>

<items>

<MenuItem fx:id=**"open"** mnemonicParsing=**"false"** onAction=**"#onFileOpen"** text=**"Open"** />

<MenuItem fx:id=**"closefile"** mnemonicParsing=**"false"** onAction=**"#onFileClose"** text=**"Close"** />

</items>

</Menu>

<Menu mnemonicParsing=**"false"** text=**"Edit"** />

<Menu mnemonicParsing=**"false"** text=**"View"**>

<items>

<CheckMenuItem fx:id=**"visScore"** mnemonicParsing=**"false"** text=**"Visualize Score"** />

</items>

</Menu>

<Menu mnemonicParsing=**"false"** text=**"Help"**>

<items>

<MenuItem mnemonicParsing=**"false"** text=**"About"** />

</items>

</Menu>

</menus>

</MenuBar>

<Button id=**"start"** fx:id=**"play"** defaultButton=**"false"** disable=**"true"** layoutX=**"714.0"** layoutY=**"91.0"** mnemonicParsing=**"false"** onAction=**"#onPlay"** prefHeight=**"38.0"** prefWidth=**"155.0"** text=**"Play"** textOverrun=**"ELLIPSIS"** wrapText=**"false"** />

<Slider id=**"slidThres"** fx:id=**"slidYsize"** layoutX=**"722.0"** layoutY=**"301.0"** majorTickUnit=**"5.0"** max=**"20.0"** minorTickCount=**"4"** showTickLabels=**"true"** showTickMarks=**"true"** snapToTicks=**"true"** value=**"15.0"** />

<Slider id=**"slidThres"** fx:id=**"slidXsize"** layoutX=**"722.0"** layoutY=**"230.0"** majorTickUnit=**"5.0"** max=**"20.0"** min=**"0.0"** minorTickCount=**"4"** showTickLabels=**"true"** showTickMarks=**"true"** snapToTicks=**"true"** value=**"15.0"** />

<Label layoutX=**"777.0"** layoutY=**"285.0"** text=**"Size Y"** />

<Label layoutX=**"777.0"** layoutY=**"214.0"** text=**"Size X"** />

<AreaChart fx:id=**"chart"** alternativeColumnFillVisible=**"false"** layoutX=**"14.0"** layoutY=**"441.0"** prefHeight=**"220.0"** prefWidth=**"660.0"**>

<xAxis>

<NumberAxis animated=**"true"** label=**"Time (s)"** side=**"BOTTOM"** tickUnit=**"100.0"** upperBound=**"500.0"** />

</xAxis>

<yAxis>

<NumberAxis label=**"Mobility"** side=**"LEFT"** tickUnit=**"5.0"** upperBound=**"15.0"** />

</yAxis>

</AreaChart>

<Label layoutX=**"708.0"** layoutY=**"506.0"** text=**"Log Data :"** />

<Label layoutX=**"717.0"** layoutY=**"551.0"** text=**"Subject :"** />

<ComboBox fx:id=**"choiceSub"** disable=**"true"** editable=**"false"** layoutX=**"785.0"** layoutY=**"551.0"** onAction=**"#onPlotDataChanged"** value=**"0"**>

<items>

<FXCollections fx:factory=**"observableArrayList"**>

<String fx:value=**"0"** />

</FXCollections>

</items>

</ComboBox>

<ComboBox fx:id=**"choiceLog"** disable=**"true"** layoutX=**"780.0"** layoutY=**"506.0"** value=**"Timeline"**>

<items>

<FXCollections fx:factory=**"observableArrayList"**>

<String fx:value=**"Stats"** />

<String fx:value=**"Timeline"** />

</FXCollections>

</items>

</ComboBox>

<Button id=**"start"** fx:id=**"clear"** defaultButton=**"false"** disable=**"false"** layoutX=**"714.0"** layoutY=**"615.0"** mnemonicParsing=**"false"** onAction=**"#onClearSubjects"** prefHeight=**"38.0"** prefWidth=**"155.0"** styleClass=**"indicator2"** text=**"Clear Subjects"** textOverrun=**"ELLIPSIS"** wrapText=**"false"** />

<TextField fx:id=**"logPath"** layoutX=**"119.0"** layoutY=**"679.0"** prefWidth=**"743.0"** />

<Label layoutX=**"48.0"** layoutY=**"682.0"** text=**"Log Path :"** />

</children>

</AnchorPane>

</children>

<columnConstraints>

<ColumnConstraints hgrow=**"SOMETIMES"** minWidth=**"-1.0"** />

</columnConstraints>

<rowConstraints>

<RowConstraints minHeight=**"10.0"** vgrow=**"SOMETIMES"** />

</rowConstraints>

<stylesheets>

<URL value=**"@styles.css"** />

</stylesheets>

</GridPane>

## styles.css

**.**root **{**

-fx-background-image**: url(**"background.jpg"**);**

**}**

**.**text-area **{**

-fx-font-size**: 12px;**

-fx-font-weight**: bold;**

-fx-text-fill**: #202f33;**

-fx-effect**: dropshadow( gaussian , rgba(255,255,255,0.5) , 0,0,0,1 );**

**}**

**.**label **{**

-fx-font-size**: 12px;**

-fx-font-weight**: bold;**

-fx-text-fill**: #202f33;**

-fx-effect**: dropshadow( gaussian , rgba(255,255,255,0.5) , 0,0,0,1 );**

**}**

**.**button **{**

-fx-font-size**: 16px;**

-fx-text-fill**: white;**

-fx-font-family**:** "Arial"**;**

-fx-font-weight**: bold;**

-fx-background-color**: linear-gradient(#61a2b1, #2A5058);**

-fx-effect**: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );**

**}**

**.**button**:hover** **{**

-fx-background-color**: linear-gradient(#2A5058, #61a2b1);**

**}**

**.**box **{**

-fx-background-color**: linear-gradient(#c5c5c5, #fcf9f6);**

-fx-effect**: dropshadow( three-pass-box , rgba(41, 41, 41, 0.6), 5, 0.0 , 0 , 1 );**

-fx-spacing**: 10;**

**}**

**.**indicator1 **{**

-fx-font-size**: 16px;**

-fx-text-fill**: white;**

-fx-font-family**:** "Arial"**;**

-fx-font-weight**: bold;**

-fx-background-color**: linear-gradient(#42b14a, #213858);**

-fx-effect**: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );**

**}**

**.**indicator1**:hover** **{**

-fx-font-size**: 16px;**

-fx-text-fill**: white;**

-fx-font-family**:** "Arial"**;**

-fx-font-weight**: bold;**

-fx-background-color**: linear-gradient(#42b14a, #213858);**

-fx-effect**: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );**

**}**

**.**indicator2 **{**

-fx-font-size**: 16px;**

-fx-text-fill**: white;**

-fx-font-family**:** "Arial"**;**

-fx-font-weight**: bold;**

-fx-background-color**: linear-gradient(#b12841, #213858);**

-fx-effect**: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );**

**}**

**.**indicator2**:hover** **{**

-fx-font-size**: 16px;**

-fx-text-fill**: white;**

-fx-font-family**:** "Arial"**;**

-fx-font-weight**: bold;**

-fx-background-color**: linear-gradient(#213858, #b12841);**

-fx-effect**: dropshadow( three-pass-box , rgba(0,0,0,0.6) , 5, 0.0 , 0 , 1 );**

**}**