



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

"Αναγνώριση μορφολογικών χαρακτηριστικών ψαριών από υποβρύχιες φωτογραφίες"

**ΣΠΗΛΙΟΣ ΚΩΣΤΟΠΟΥΛΟΣ**

ΕΠΙΒΛΕΠΩΝ: ΝΙΚΟΛΑΟΣ ΠΕΤΡΕΛΛΗΣ, Αναπληρωτής Καθηγητής

ΠΑΤΡΑ 2023

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή

Πάτρα, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Νικόλαος Πετρέλλης, Υπογραφή
2. Νικόλαος Βώρος, Υπογραφή
3. Χρήστος Αντωνόπουλος, Υπογραφή

## **Υπεύθυνη Δήλωση Φοιτητή**

*Βεβαιώνω ότι είμαι συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τη συγκεκριμένη εργασία.*

*Η έγκριση της διπλωματικής εργασίας από το Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Πανεπιστημίου Πελοποννήσου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Σπήλιου Κωστόπουλου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο Πανεπιστήμιο Πελοποννήσου, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.*

## **ΠΕΡΙΛΗΨΗ**

Η συγκεκριμένη μελέτη εστιάζει στον εντοπισμό σημείων (landmarks) σε εικόνες ψαριών, χρησιμοποιώντας μια σειρά από μοντέλα. Προσεγγίζονται διάφοροι παράγοντες όπως ο αριθμός των σταδίων (cascade stages) και των δέντρων παλινδρόμησης (ERT) για τη βελτιστοποίηση των αποτελεσμάτων. Η ανάλυση απεικονίζει τη σύγκριση των αποτελεσμάτων κάθε μοντέλου με τις αναφορικές συντεταγμένες των σημείων που έχουν χειροκίνητα τοποθετηθεί. Επίσης γίνεται περιγραφή της εφαρμογής και του κώδικα που χρησιμοποιήθηκε.

## ΠΕΡΙΕΧΟΜΕΝΑ

1. ΕΙΣΑΓΩΓΗ .....	6
1.1. ΜΕΘΟΔΟΛΟΓΙΑ .....	6
1.2. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΜΕΘΟΔΟΥ ΣΤΟΙΧΙΣΗΣ ΠΡΟΣΩΠΟΥ ΤΩΝ ΚΑΖΕΜΙ & SULLIVAN .....	7
2. ΜΕΘΟΔΟΙ ΥΠΟΒΡΥΧΙΑΣ ΠΑΡΑΤΗΡΗΣΗΣ ΨΑΡΙΩΝ .....	9
3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ VISUAL STUDIO .....	11
4. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ ΤΩΝ ΕΦΑΡΜΟΓΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ .....	17
4.1. DEST_ALIGN .....	17
4.2. DEST_GENERATE_RECTANGLES .....	19
4.3. DEST_TRAIN .....	20
5. ΣΤΟΙΧΕΙΑ ΚΑΙ ΔΕΔΟΜΕΝΑ .....	21
6. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ .....	22
7. ΒΙΒΛΙΟΓΡΑΦΙΑ .....	39

## **1. ΕΙΣΑΓΩΓΗ**

Η παρούσα έρευνα αποτελείται από σημαντικά ευρήματα σχετικά με τον εντοπισμό σημείων (landmarks) σε εικόνες ψαριών, παρέχοντας πληροφορίες για την αποτελεσματικότητα διαφόρων παραμέτρων πάνω στην ακρίβεια και την απόδοση των μοντέλων δέντρων παλινδρόμησης (ERT). Τα ευρήματα αυτά μπορούν να συμβάλουν στην περαιτέρω βελτίωση και ανάπτυξη τεχνικών ανίχνευσης σημείων (landmarks) σε εικόνες με πρακτικές εφαρμογές σε διάφορους τομείς όπως η υγειονομική φροντίδα, η υπολογιστική όραση και η ασφάλεια.

Αναλυτικότερα η μελέτη περιλαμβάνει βιβλιογραφική αναφορά διαφόρων εφαρμογών και μεθόδων υποβρύχιας παρακολούθησης και καταγραφής δεδομένων όπως επίσης και μια σειρά από μοντέλα που αναπτύχθηκαν για τον εντοπισμό σημείων (landmarks) σε εικόνες ψαριών. Η εκπαίδευση του πρώτου μοντέλου περιλάμβανε 319 εικόνες, με την χρήση μιας αλληλουχίας δέκα (10) σταδίων (cascade stages) και πεντακοσίων (500) δέντρων παλινδρόμησης (ERT). Για τον έλεγχο της ακρίβειας, χρησιμοποιήθηκαν 50 εικόνες με 18 σημεία (landmarks) που τοποθετήθηκαν χειροκίνητα ως σημεία αναφοράς.

Προκειμένου να διερευνηθεί η επίδραση διαφορετικών παραμέτρων στην ακρίβεια και τον χρόνο εκπαίδευσης, δημιουργήθηκαν άλλα 6 μοντέλα. Κάθε μοντέλο διαφοροποιείται στον αριθμό των σταδίων αλληλουχίας (cascade stages) και των δέντρων παλινδρόμησης (ERT). Τα αποτελέσματα των μοντέλων αξιολογήθηκαν με τη χρήση των 50 εικόνων ελέγχου και με τη σύγκριση των προβλεπόμενων σημείων (landmarks) με τα χειροκίνητα τοποθετημένα σημεία αναφοράς.

Το αποτέλεσμα της ανάλυσης παρουσιάζεται μέσω διαγραμμάτων που απεικονίζουν τη σύγκριση των αποτελεσμάτων κάθε μοντέλου με τις χειροκίνητα τοποθετημένες αναφορές. Συγκρίνονται οι τιμές των συντεταγμένων των σημείων που προβλέπονται από τα μοντέλα με τις αντίστοιχες τιμές των αναφορικών σημείων, προκειμένου να εκτιμηθεί η ακρίβεια και η απόδοση της μεθόδου σε κάθε περίπτωση.

### **1.1. ΜΕΘΟΔΟΛΟΓΙΑ**

Για την εκπόνηση της εργασίας αρχικά μελετήθηκε το υπάρχον θεωρητικό υπόβαθρο τόσο των μεθόδων παρατήρησης ψαριών υποβρύχια όσο και για το τι εφαρμογές έχουν προταθεί και

συγκεντρώθηκε η κατάλληλη βιβλιογραφία για τη δημιουργία του θεωρητικού μέρους. Στη συνέχεια ακολούθησε η συλλογή των στοιχείων για την περιοχή της μελέτης τα οποία περιλάμβαναν: Εκπαίδευση των μοντέλων, διεξαγωγή μετρήσεων και συγκέντρωση των δεδομένων.

## **1.2. ΠΕΡΙΓΡΑΦΗ ΤΗΣ ΜΕΘΟΔΟΥ ΣΤΟΙΧΙΣΗΣ ΠΡΟΣΩΠΟΥ ΤΩΝ KAZEMI & SULLIVAN**

Η μέθοδος στοίχισης σχήματος προσώπου που παρουσιάζουν οι Kazemi και Sullivan βασίζεται σε ένα σύνολο δέντρων παλινδρόμησης (Ensemble of Regression Trees - ERT) για την ακριβή ευθυγράμμιση των πολυάριθμων σημείων προσώπου σε μια εικόνα. Η μέθοδος αυτή έχει αποδειχθεί ότι επιτυγχάνει ακρίβεια και ταχύτητα στην εκτέλεσή της, καθιστώντας την κατάλληλη για εφαρμογές πραγματικού χρόνου.

Στη μέθοδο αυτή, αρχικά εκπαιδεύονται τα δέντρα παλινδρόμησης χρησιμοποιώντας ένα σύνολο εκπαίδευσης που περιλαμβάνει εικόνες προσώπων και τις αντίστοιχες θέσεις των σημείων προσώπου σε αυτές τις εικόνες. Κατά τη διάρκεια της εκπαίδευσης, τα δέντρα παλινδρόμησης εκτιμούν τις θέσεις των σημείων προσώπου βάσει των χαρακτηριστικών που περιγράφουν την εικόνα. Κατόπιν, οι εκτιμήσεις από όλα τα δέντρα συνδυάζονται για να παράξουν την τελική ευθυγράμμιση του προσώπου.

Μια σημαντική πτυχή της μεθόδου είναι η ταχύτητά της εκτέλεσης. Η χρήση του ERT επιτρέπει την παράλληλη εκτέλεση των δέντρων, επιταχύνοντας την διαδικασία της ευθυγράμμισης. Επιπλέον, η μέθοδος χρησιμοποιεί αποδοτικά χαρακτηριστικά που περιγράφουν το πρόσωπο και επιτρέπουν την γρήγορη και ακριβή εκτίμηση των θέσεων των σημείων προσώπου.

Η μέθοδος αυτή έχει εφαρμογές σε πολλούς τομείς, όπως η αναγνώριση προσώπου, η ανίχνευση εκφράσεων προσώπου και η αλληλεπίδραση υπολογιστή-ανθρώπου. Μπορεί να χρησιμοποιηθεί για την ανίχνευση και παρακολούθηση της κατάστασης ενός οδηγού σε οχήματα, την αναγνώριση και απόδοση εκφράσεων προσώπου για συστήματα ανθρώπινης αλληλεπίδρασης, καθώς και για την ανάλυση και αξιολόγηση των προσώπων σε εικόνες και βίντεο.

Συνολικά, η μέθοδος στοίχισης σχήματος προσώπου των Kazemi και Sullivan είναι μια προηγμένη τεχνική που επιτυγχάνει ακρίβεια και ταχύτητα στην ευθυγράμμιση των πολυάριθμων σημείων προσώπου σε μια εικόνα. Η εφαρμογή της σε διάφορους τομείς επιτρέπει την ανάπτυξη

προηγμένων εφαρμογών που απαιτούν αναγνώριση και ανάλυση του προσώπου, ενισχύοντας την αλληλεπίδραση ανθρώπου-υπολογιστή και την αναγνώριση συναισθημάτων σε πραγματικό χρόνο.



## **2. ΜΕΘΟΔΟΙ ΥΠΟΒΡΥΧΙΑΣ ΠΑΡΑΤΗΡΗΣΗΣ ΨΑΡΙΩΝ**

Η υποβρύχια παρατήρηση ψαριών είναι ένας σημαντικός τομέας της θαλάσσιας επιστήμης και ιχθυολογίας. Ο σκοπός αυτής της αναζήτησης είναι να κατανοήσουμε και να παρακολουθήσουμε τη συμπεριφορά, την κίνηση, τη διατήρηση και τις παραμέτρους περιβάλλοντος των ψαριών στον υποβρύχιο χώρο. Με τη βοήθεια της τεχνολογίας, έχουν αναπτυχθεί πολλές μέθοδοι παρατήρησης που επιτρέπουν τη συλλογή ακριβών και λεπτομερών δεδομένων για τα ψάρια σε πραγματικό χρόνο.

Στο πλαίσιο αυτού του κεφαλαίου, θα αναλύσουμε μερικές από τις σημαντικότερες μεθόδους παρατήρησης που έχουν προταθεί στη βιβλιογραφία και θα εξετάσουμε τις παραμέτρους που παρατηρούνται κατά την εφαρμογή τους.

Σύμφωνα με τους Chen et al. (2022) παρέχεται μια συνοπτική επισκόπηση των τεχνικών παρατήρησης υποβρυχίων πληθυσμών ψαριών, αναλύοντας τις εφαρμογές τους και παρουσιάζοντας μελλοντικές προοπτικές για το πεδίο. Οι συγγραφείς εξετάζουν διάφορες τεχνικές παρατήρησης, συμπεριλαμβανομένης της χρήσης ψηφιακής κάμερας και τεχνικών όρασης υπολογιστή, καθώς και την ακουστική παρατήρηση.

Οι Alexandridis et al. (2019) παρουσιάζουν την εφαρμογή ψηφιακής κάμερας και τεχνικών όρασης υπολογιστών για την ανίχνευση ψαριών υποβρυχίως και περιγράφουν διάφορες μεθόδους επεξεργασίας εικόνας και αναλύουν τα αποτελέσματα που επιτεύχθηκαν με αυτές τις μεθόδους.

Επιπλέον, η αναγνώριση ειδών ψαριών μπορεί να γίνει μέσω της χρήσης αλγορίθμων μηχανικής μάθησης (Sun et al., 2020). Σε αυτήν τη μελέτη, προτείνεται μια προσέγγιση αναγνώρισης ψαριών υποβρυχίως με βάση την οπτική εμφάνιση και το συνελκτικό νευρωνικό δίκτυο και εξετάζεται η απόδοση της μεθόδου τους σε πραγματικά δεδομένα υποβρύχιας παρατήρησης.

Οι Morais et al. (2013) σε αυτήν τη μελέτη, παρουσιάζουν διάφορες μεθόδους παρατήρησης ψαριών υποβρυχίως και αναλύονται οι πιθανοί παράγοντες παραμόρφωσης και υποβάθμισης των δεδομένων που προκύπτουν από αυτές τις μεθόδους.

Επιπλέον προτείνεται ένα σύστημα αναγνώρισης ειδών ψαριών υποβρυχίως βασισμένο σε βαθιά μάθηση με επαυξημένα δεδομένα (Huang et al., 2021). Οι συγγραφείς παρουσιάζουν ακριβείς αποτελέσματα αναγνώρισης ειδών ψαριών με χρήση αυτής της μεθόδου.

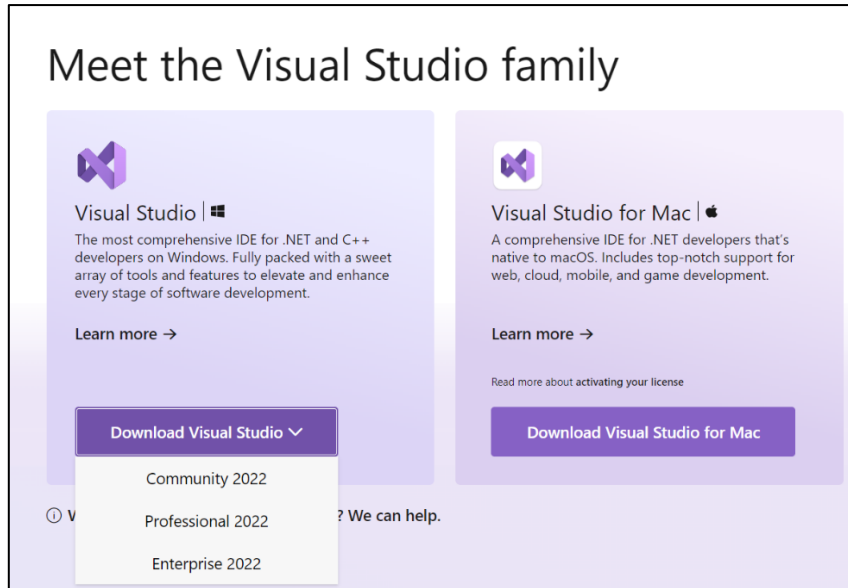
Μία τελευταία αναφορά αλλά εξίσου σημαντική με τις προηγούμενες αποτελούν οι προηγμένες τεχνικές υποβρυχίας ακουστικής παρατήρησης για τη μελέτη των πληθυσμών ψαριών όπου για την υλοποίησή τους γίνεται η χρήση υποβρυχίων ακουστικών συστημάτων για τη συλλογή αξιόπιστων και με μεγάλη ακρίβεια δεδομένων για τα ψάρια (Smith et al., 2022).

Με βάση τις παραπάνω πηγές, μπορούμε να αναλύσουμε τις διάφορες μεθόδους παρατήρησης ψαριών υποβρυχίως και τις συναφείς παραμέτρους που παρατηρούνται κατά την εφαρμογή τους.

### **3. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΕΡΙΒΑΛΛΟΝΤΟΣ VISUAL STUDIO**

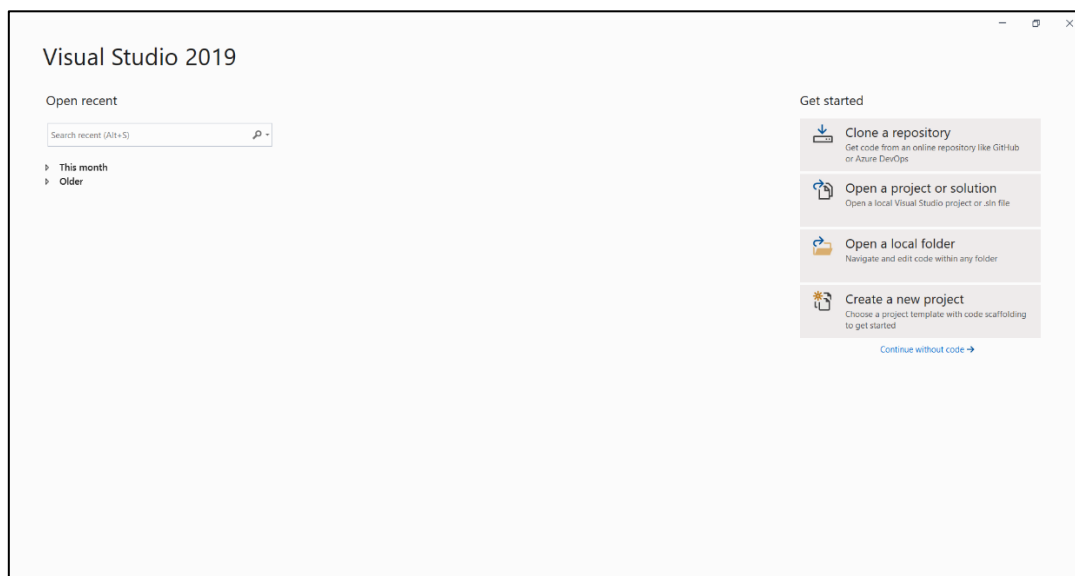
Το περιβάλλον του Visual Studio είναι μια ολοκληρωμένη σουίτα ανάπτυξης λογισμικού που παρέχει ένα ευέλικτο και πανίσχυρο περιβάλλον για προγραμματιστές. Με το Visual Studio, μπορούμε να αναπτύξουμε διάφορους τύπους εφαρμογών, όπως εφαρμογές για Windows, ιστοσελίδες, εφαρμογές κινητών συσκευών και ακόμα και εφαρμογές τεχνητής νοημοσύνης. Το Visual Studio υποστηρίζει πολλές γλώσσες προγραμματισμού, όπως C++, C#, Visual Basic, Python, JavaScript και πολλές άλλες. Επιπλέον, παρέχει ευέλικτες δυνατότητες ενσωμάτωσης και ανάπτυξης ιστοσελίδων με τη χρήση του ASP.NET και άλλων τεχνολογιών. Το Visual Studio διαθέτει επίσης πολλά εργαλεία για τη διαχείριση του κώδικα, την ομαδική εργασία και τη δοκιμή της εφαρμογής. Μπορούμε να χρησιμοποιήσουμε το Git για τον έλεγχο των εκδόσεων του κώδικα, να συνεργαστούμε με άλλους προγραμματιστές μέσω ενσωματωμένων εργαλείων συνεργασίας και να δοκιμάσουμε την εφαρμογή μας σε διάφορα περιβάλλοντα και συσκευές. Επιπλέον, το Visual Studio παρέχει πρόσβαση σε πλούσια τεκμηρίωση και πόρους, καθώς και στην παγκόσμια κοινότητα των προγραμματιστών. Μπορούμε να βρούμε άρθρα, εκπαιδευτικά βίντεο, παραδείγματα κώδικα και άλλους πόρους που θα μας βοηθήσουν να μάθουμε και να βελτιώσουμε τις δεξιότητές μας στον προγραμματισμό.

Ξεκινώντας, η εγκατάσταση του συγκεκριμένου προγράμματος είναι πολύ απλή και το μόνο που χρειάζεται είναι να κατευθυνθεί ο ενδιαφερόμενος στην ιστοσελίδα της Microsoft και να επιλέξει την κατάλληλη έκδοση ώστε να είναι συμβατή με το σύστημα του (Windows – Mac) όπως επίσης και τις απαιτήσεις του είτε αυτές είναι επαγγελματικές είτε σε επίπεδο εταιρείας (professional – enterprise) και να πατήσει το κουμπί «λήψη».

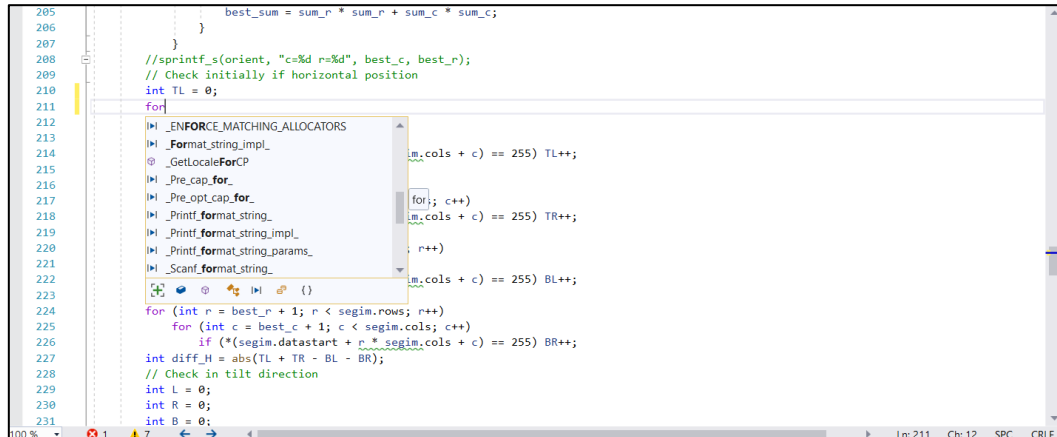


Το περιβάλλον του Visual Studio παρέχει μια ολοκληρωμένη σουίτα εργαλείων για τη δημιουργία, την επεξεργασία και τον έλεγχο του κώδικα. Εδώ είναι μερικά ενδεικτικά στιγμιότυπα οθόνης που δείχνουν το περιβάλλον του Visual Studio:

1. Αρχική οθόνη: Μετά την εκκίνηση του Visual Studio, θα συναντήσουμε την αρχική οθόνη όπου μπορούμε να δημιουργήσουμε ένα νέο έργο, να ανοίξουμε ένα υπάρχον έργο, ή να προσαρμόσουμε τις προτιμήσεις μας.

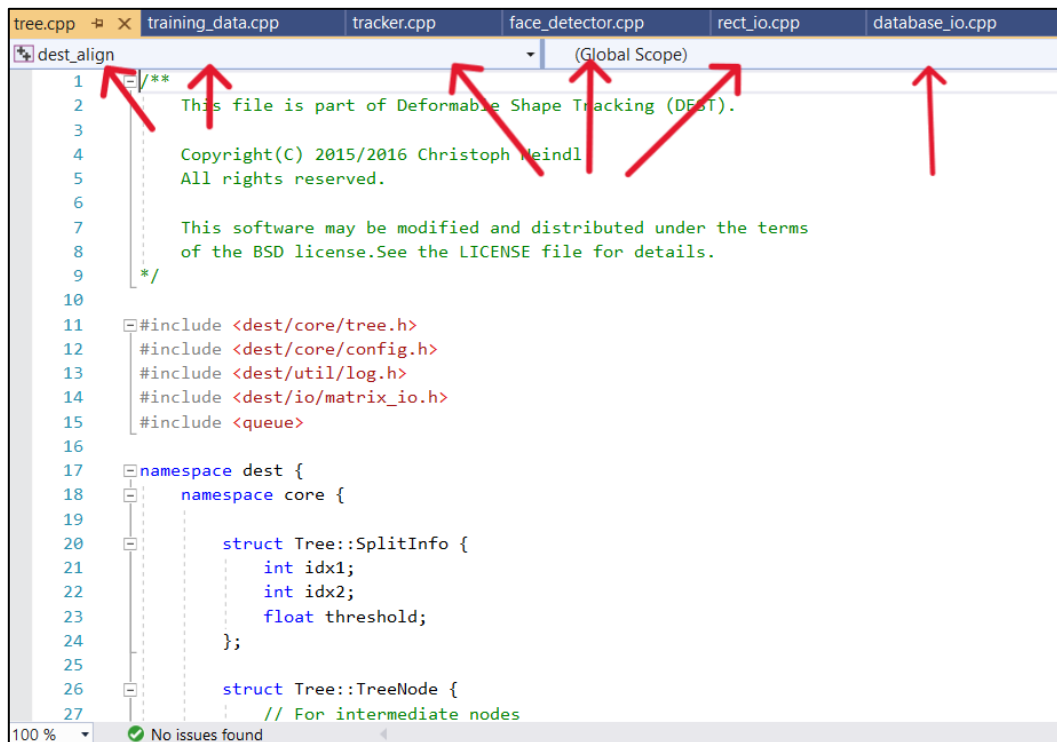


- Επεξεργαστής κειμένου: Το Visual Studio παρέχει έναν προηγμένο επεξεργαστή κειμένου με χαρακτηριστικά όπως επισήμανση σύνταξης και η αυτόματη συμπλήρωση κώδικα. Μπορούμε να εργαστούμε σε αρχεία κώδικα C++, C#, Visual Basic και πολλές άλλες γλώσσες προγραμματισμού.



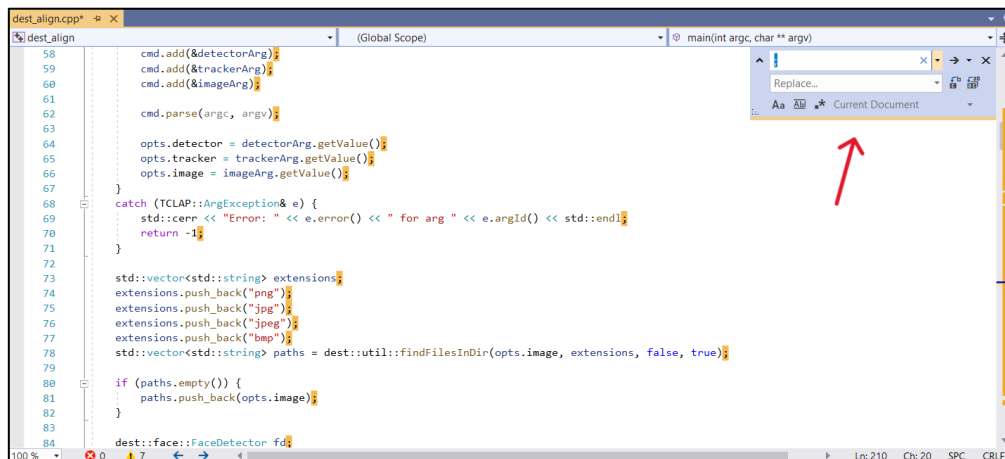
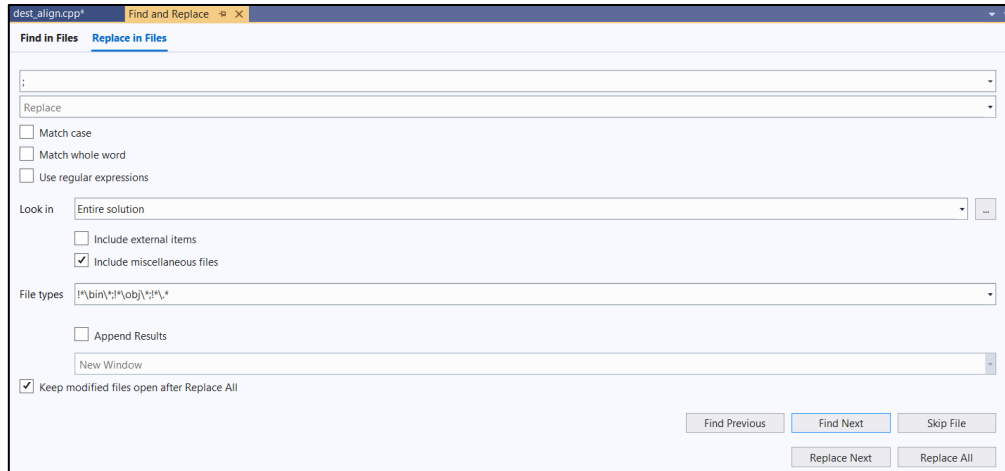
```
205         best_sum = sum_r * sum_r + sum_c * sum_c;
206     }
207 }
208 //sprintf_s(orient, "c=%d r=%d", best_c, best_r);
209 // Check initially if horizontal position
210 int TL = 0;
211 for
212     |<_ENFORCE_MATCHING_ALLOCATORS
213     |<_Format_string_impl_
214     |<_GetLocalForCP
215     |<_Pre_cap_for_
216     |<_Pre_opt_cap_for_
217     |<_Print_format_string_
218     |<_Print_format_string_impl_
219     |<_Print_format_string_params_
220     |<_Scanf_format_string_
221
222     |<_m_cols + c) == 255) TL++;
223
224     |<for; c++)
225     |<_m_cols + c) == 255) TR++;
226     |<r++)
227     |<_m_cols + c) == 255) BL++;
228
229 for (int r = best_r + 1; r < segim.rows; r++)
230     for (int c = best_c + 1; c < segim.cols; c++)
231         if (*(segim.datastart + r * segim.cols + c) == 255) BR++;
232
233 int diff_H = abs(TL + TR - BL - BR);
234 // Check in tilt direction
235 int L = 0;
236 int R = 0;
237 int B = 0;
```

- Ανακατατάξιμες καρτέλες: Το Visual Studio μας επιτρέπει να εργαστούμε με πολλά αρχεία κώδικα ταυτόχρονα, τοποθετώντας τα σε ανακατατάξιμες καρτέλες. Αυτό μας επιτρέπει να πλοηγηθούμε εύκολα μεταξύ των αρχείων και να συγκρίνουμε τον κώδικα.

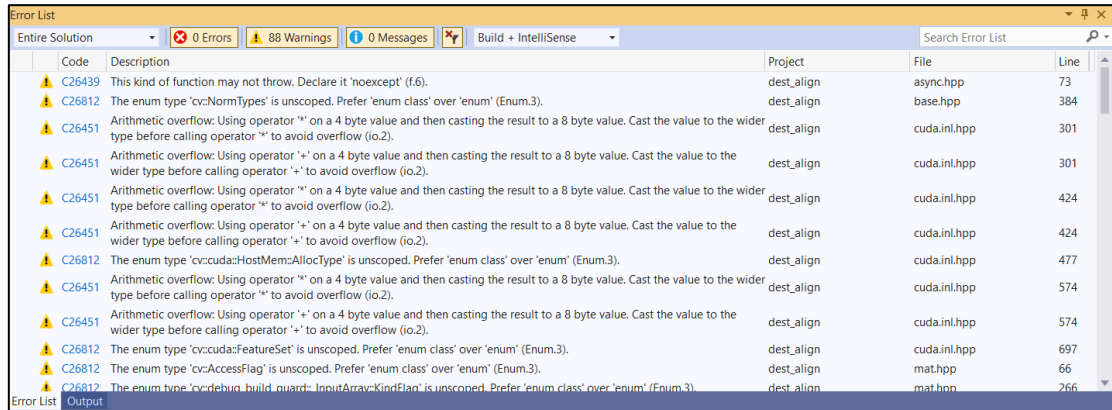


```
tree.cpp x training_data.cpp tracker.cpp face_detector.cpp rect_io.cpp database_io.cpp
dest_align (Global Scope)
1 /**
2 This file is part of Deformable Shape Tracking (DST).
3
4 Copyright(C) 2015/2016 Christoph Heindl
5 All rights reserved.
6
7 This software may be modified and distributed under the terms
8 of the BSD license. See the LICENSE file for details.
9 */
10
11 #include <dest/core/tree.h>
12 #include <dest/core/config.h>
13 #include <dest/util/log.h>
14 #include <dest/io/matrix_io.h>
15 #include <queue>
16
17 namespace dest {
18     namespace core {
19
20         struct Tree::SplitInfo {
21             int idx1;
22             int idx2;
23             float threshold;
24         };
25
26         struct Tree::TreeNode {
27             // For intermediate nodes
```

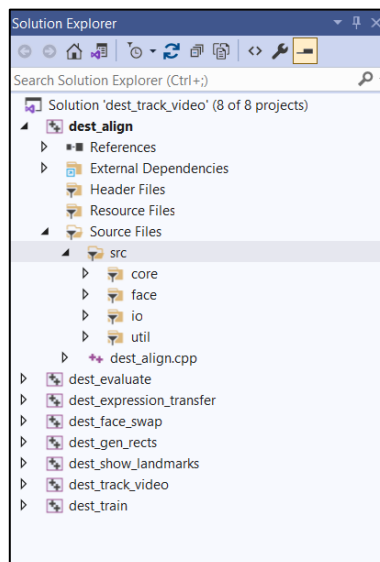
4. Αναζήτηση και αντικατάσταση: Το Visual Studio παρέχει ισχυρά εργαλεία αναζήτησης και αντικατάστασης που μας επιτρέπουν να βρούμε γρήγορα και εύκολα κομμάτια κώδικα και να τα τροποποιήσουμε.



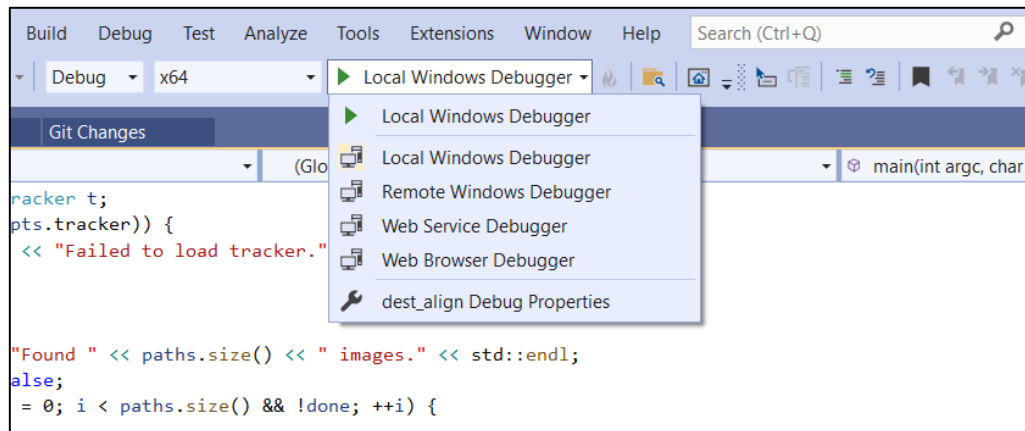
5. Αντιμετώπιση σφαλμάτων: Το Visual Studio παρέχει έναν αναλυτή κώδικα που επιτρέπει τον εντοπισμό τυπικών λαθών, την πρόταση βελτιώσεων κώδικα και την αυτόματη διόρθωση σφαλμάτων. Αυτό μας βοηθά να εξοικονομήσουμε χρόνο και προσπάθεια κατά τη διάρκεια της ανάπτυξης του λογισμικού μας.



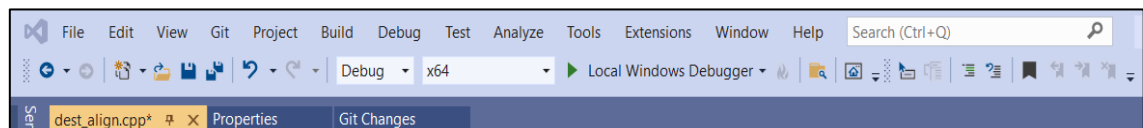
6. Το Solution Explorer παρουσιάζει μια οργανωμένη ιεραρχία όλων των αρχείων που ανήκουν σε ένα έργο λογισμικού (project) και της δομής του ίδιου του έργου. Με τη βοήθεια του Solution Explorer, μπορείς να περιηγηθείς στα αρχεία του έργου σου, να τα οργανώσεις σε φακέλους, να τα ανοίξεις για επεξεργασία, να τους αλλάξεις όνομα, να τα διαγράψεις και να τα επανατοποθετήσεις στη δομή του έργου. Επίσης, μπορείς να δημιουργήσεις νέα αρχεία ή φακέλους, να προσθέσεις υπάρχοντα αρχεία στο έργο σου και να διαχειριστείς τις εξαρτήσεις μεταξύ των αρχείων.



7. Debugger: Ένα εργαλείο παρακολούθησης και ανάλυσης της εκτέλεσης του προγράμματος σου, που σου επιτρέπει να εντοπίσεις και να επιδιορθώσεις σφάλματα και προβλήματα.



8. Εργαλεία διαχείρισης έργου: Παρέχουν δυνατότητες για διαχείριση της έκδοσης του λογισμικού, διαμοιρασμό κώδικα με άλλους προγραμματιστές, καταγραφή αλλαγών και παρακολούθηση των αναφορών σφαλμάτων.



Με αυτές και ακόμα περισσότερες δυνατότητες, το περιβάλλον του Visual Studio είναι ένα ισχυρό εργαλείο για προγραμματιστές και ομάδες ανάπτυξης που αναζητούν ένα ολοκληρωμένο και αποδοτικό περιβάλλον για τη δημιουργία υψηλής ποιότητας λογισμικού.



## **4. ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΚΩΔΙΚΑ ΤΩΝ ΕΦΑΡΜΟΓΩΝ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ**

### **4.1. DEST\_ALIGN**

Ο κώδικας περιλαμβάνει διάφορα αρχεία που απαιτούνται για την επεξεργασία εικόνας, την ανίχνευση προσώπου και την παρακολούθηση με τη χρήση της βιβλιοθήκης DEST. Η main είναι το σημείο εισόδου του προγράμματος. Αναλύει τα ορίσματα της γραμμής εντολών για το αρχείο εντοπισμού (tracker), το αρχείο ανίχνευσης (detector) και την εικόνα ή τις εικόνες προς επεξεργασία. Στη συνέχεια, το πρόγραμμα αναζητά αρχεία εικόνων με βάση την παρεχόμενη είσοδο. Εάν παρέχεται ένας κατάλογος, αναζητά αρχεία εικόνας με συγκεκριμένες επεκτάσεις (png, jpg, jpeg, bmp) εντός του καταλόγου. Ο ανιχνευτής προσώπου OpenCV αρχικοποιείται και φορτώνεται με το καθορισμένο αρχείο detector όπως αντίστοιχα και ο DEST tracker με το καθορισμένο αρχείο tracker. Το πρόγραμμα εκτελώντας βρόχο σε κάθε αρχείο εικόνας εκτελεί τα ακόλουθα βήματα:

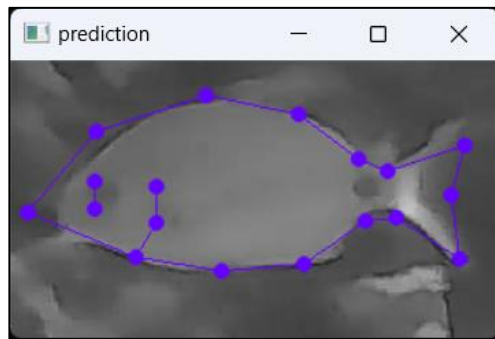
1. Διαβάζει την εικόνα χρησιμοποιώντας τη συνάρτηση imread του OpenCV.
2. Μετατρέπει την εικόνα σε κλίμακα του γκρι και δημιουργεί ένα αντικείμενο (dest::core::Image) για επεξεργασία με το DEST.
3. Προ-επεξεργάζεται (preprocessing) την εικόνα εφαρμόζοντας λειτουργίες οριοθέτησης και θόλωσης.
4. Υπολογίζει τον προσανατολισμό της εικόνας χρησιμοποιώντας μια μέθοδο που βασίζεται στο κέντρο βάρους και στην καταμέτρηση των εικονοστοιχείων.
5. Εμφανίζει την προ-επεξεργασμένη (pre-processed) εικόνα με μια ετικέτα προσανατολισμού χρησιμοποιώντας τη συνάρτηση imshow του OpenCV.



6. Προβλέπει τα σημεία (landmarks) του σχήματος χρησιμοποιώντας τον φορτωμένο ανιχνευτή (tracker) και την προ-επεξεργασμένη (pre-processed) εικόνα.



7. Ενημερώνει την εικόνα με τα προβλεπόμενα σημεία (landmarks) και την εμφανίζει χρησιμοποιώντας τη συνάρτηση `imshow` του OpenCV.



8. Τέλος δημιουργεί ένα αρχείο `.txt` στο οποίο γράφει τις συντεταγμένες των σημείων (landmarks) που δημιούργησε.

## 4.2. DEST\_GENERATE\_RECTANGLES

Ο κώδικας παράγει ορθογώνια οριοθέτησης για την εκπαίδευση του ανιχνευτή (tracker) χρησιμοποιώντας τον ανιχνευτή προσώπου OpenCV. Λαμβάνει μια βάση δεδομένων προσώπων/σχημάτων ως είσοδο και εξάγει τα ορθογώνια σε ένα αρχείο CSV για την εκπαίδευση του ανιχνευτή (tracker). Ακολουθεί μια ανάλυση του κώδικα:

1. Το πρόγραμμα περιλαμβάνει έναν μεγάλο όγκο αρχείων για διάφορες εξαρτήσεις (dependencies), όπως το OpenCV, το TCLAP (Templatized C++ Command Line Parser Library) και προσαρμοσμένα αρχεία βιβλιοθηκών από το χώρο ονομάτων (namespace) "dest".
2. Η συνάρτηση `ratioRectShapeOverlap` υπολογίζει την επικάλυψη μεταξύ ενός ορθογωνίου ( $r$ ) και ενός σχήματος ( $s$ ) χρησιμοποιώντας διανύσματα Eigen και πίνακες.
3. Η κύρια συνάρτηση (`main`) είναι το σημείο εισόδου του προγράμματος.
4. Το πρόγραμμα ορίζει έναν αριθμό από `FallbackMode` για να αναπαριστά τα διαφορετικά `fallback modes` όταν ο ανιχνευτής OpenCV αποτυγχάνει.
5. Το πρόγραμμα ορίζει μια δομή `opts` για την αποθήκευση των επιλογών γραμμής εντολών που παρέχει ο χρήστης.
6. Το πρόγραμμα χρησιμοποιεί τη βιβλιοθήκη TCLAP για να αναλύει τα ορίσματα της γραμμής εντολών και να τα αποθηκεύει στη δομή `opts`.
7. Το πρόγραμμα ρυθμίζει τις απαραίτητες ρυθμίσεις για τη χρήση του OpenCV, συμπεριλαμβανομένης της απενεργοποίησης του OpenCL αν η έκδοση του OpenCV είναι 3.0.
8. Δημιουργείται ένα αντικείμενο `ShapeDatabase (sd)` για τη φόρτωση της βάσης δεδομένων προσώπων/σχημάτων.
9. Το πρόγραμμα φορτώνει τις εικόνες, τα σχήματα και τα ορθογώνια από τη βάση δεδομένων χρησιμοποιώντας τη μέθοδο `load` της `ShapeDatabase`.
10. Οι ανιχνευτές (detectors) προσώπων αρχικοποιούνται και φορτώνονται με ταξινομητές (classifiers) που καθορίζονται από τον χρήστη.
11. Ένας βρόχος επαναλαμβάνει τις εικόνες στη βάση δεδομένων και εκτελεί την ανίχνευση προσώπων χρησιμοποιώντας τους φορτωμένους ανιχνευτές (detectors).
12. Το πρόγραμμα καθορίζει το καλύτερο ορθογώνιο προσώπου με βάση την επικάλυψη σχήματος και τον καθορισμένο `fallback mode` σε περίπτωση αποτυχίας της ανίχνευσης.

13. Παρακολουθείται ο αριθμός των επιτυχημένων ανιχνεύσεων.
14. Τα ορθογώνια που προκύπτουν εξάγονται σε αρχείο CSV χρησιμοποιώντας τη συνάρτηση `exportRectangles` από το χώρο ονομάτων (namespace) `io`.
15. Το πρόγραμμα τερματίζεται.

### **4.3. DEST\_TRAIN**

Το πρόγραμμα αυτό δέχεται ορισμένες παραμέτρους από τη γραμμή εντολών και πραγματοποιεί την εκπαίδευση ενός συστήματος παρακολούθησης προσώπου. Ο κώδικας περιλαμβάνει επίσης τη χρήση της βιβλιοθήκης `OpenCV` για την επεξεργασία εικόνας, όπως τη φόρτωση εικόνων, την εφαρμογή φίλτρων και την εμφάνιση των αποτελεσμάτων. Επιπλέον, χρησιμοποιεί άλλες βιβλιοθήκες για την εκπαίδευση του συστήματος παρακολούθησης προσώπου, όπως η `"dest"` και η `"TCLAP"`. Ο κώδικας αρχίζει με τη δήλωση και την αρχικοποίηση μιας δομής που περιέχει τις παραμέτρους του προγράμματος. Στη συνέχεια, χρησιμοποιεί τη βιβλιοθήκη `TCLAP` για τη διαχείριση των ορισμάτων γραμμής εντολών που παρέχονται στο πρόγραμμα. Στη συνέχεια, ο κώδικας φορτώνει ορισμένα δεδομένα από μια βάση δεδομένων, πραγματοποιεί επεξεργασία των δεδομένων αυτών και δημιουργεί εκπαιδευτικά δείγματα. Έπειτα, εκπαιδεύει έναν `tracker` χρησιμοποιώντας τα δείγματα αυτά και αποθηκεύει τον εκπαιδευμένο `tracker` σε ένα αρχείο εξόδου.

## 5. ΣΤΟΙΧΕΙΑ ΚΑΙ ΔΕΔΟΜΕΝΑ

Το πλήθος των φωτογραφιών που χρησιμοποιήθηκε για την διεξαγωγή της εκπαίδευσης των μοντέλων και την υλοποίηση των πειραμάτων που ήταν απαραίτητα για την εκπόνηση της πτυχιακής ανέρχεται στις 319. Το μέγεθος τους κυμαίνεται από 2 KB στην μικρότερη έως τα 541 KB την μεγαλύτερη με την μέση τιμή να είναι στα 79 KB. Η ανάλυση τους αντίστοιχα είναι από 53 x 21 pixels για την χαμηλότερη και 1134 x 585 pixels για την υψηλότερη, με την μέση ανάλυση να είναι περίπου στα 500 x 300 pixels. Ένας μικρός αριθμός από αυτές παρατίθεται στη συνέχεια:



## 6. ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ

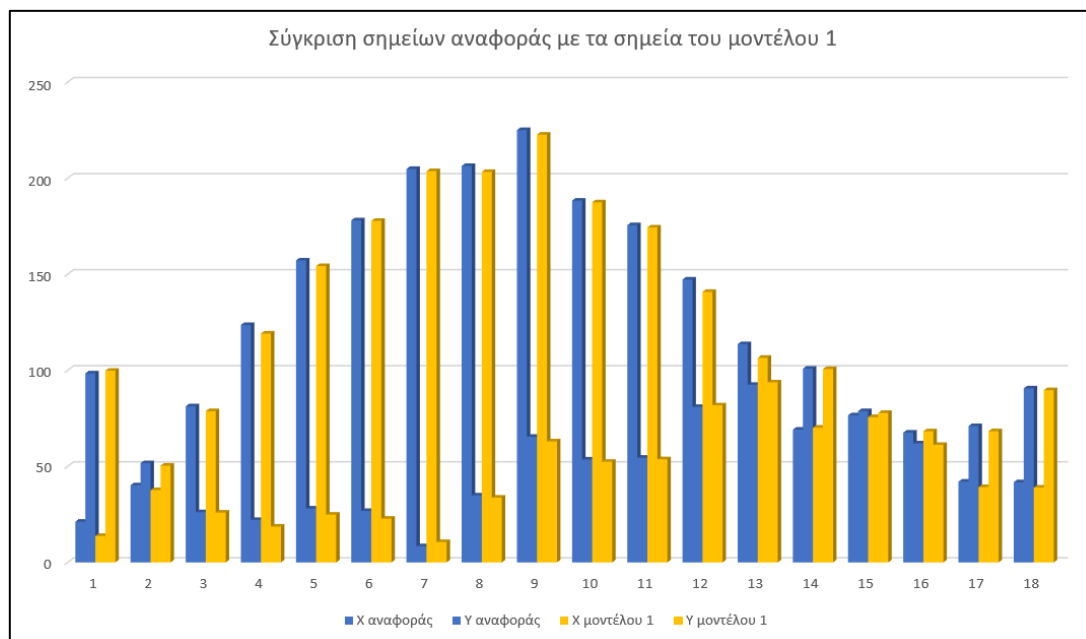
Αρχικά πραγματοποιήθηκε η εκπαίδευση του πρώτου μοντέλου που χρησιμοποιήθηκε στις μετρήσεις έχοντας ένα πλήθος 319 εικόνων, του οποίου η δομή αποτελείται από μια αλληλουχία δέκα (10) σταδίων (cascade stages) και πεντακοσίων (500) δέντρων παλινδρόμησης (ERT). Στην συνέχεια μετά την εκπαίδευση του παραπάνω μοντέλου χρησιμοποιήθηκαν 50 τυχαίες εικόνες για τον έλεγχο της λειτουργίας του και της ακρίβειας που διαθέτει. Για κάθε μια από αυτές έγινε η τοποθέτηση των 18 σημείων (landmarks) με το χέρι ως σημείο αναφοράς και στη συνέχεια συγκρίνοντας τις συντεταγμένες των σημείων (landmarks) που προέβλεψε το μοντέλο με τις προαναφερθείσες είναι εφικτός ο εντοπισμός του σφάλματος τόσο ως προς κάθε σημείο (landmark) ξεχωριστά αλλά και ως μέσο όρο όλων των σημείων (landmarks) που μας δείχνει γενικά σαν σύνολο όλη την εικόνα. Ο υπολογισμός του μέσου όρου σφάλματος υλοποιήθηκε με δύο τρόπους, ο πρώτος είναι το απόλυτο σφάλμα και ο δεύτερος είναι το κανονικοποιημένο σφάλμα. Στις παρακάτω εικόνες των μετρήσεων (1 από τις 50 εικόνες που χρησιμοποιήθηκαν) φαίνονται τα σημεία αναφοράς που δημιουργήσαμε με το χέρι μαζί με τα αποτελέσματα του πρώτου μοντέλου και δίπλα το σφάλμα (στην πρώτη εικόνα το απόλυτο και στην δεύτερη εικόνα το κανονικοποιημένο).

	Reference		fish2_left		Error
	X	Y	X	Y	
landmark1	21,16146	98,33854	13,7421	99,7841	7,558872
landmark2	40,14453	51,65885	37,6357	50,3691	2,820937
landmark3	81,22266	26,14063	78,7319	25,8981	2,50254
landmark4	123,5456	22,09505	119,121	18,6376	5,615251
landmark5	157,1549	28,00781	154,234	24,8403	4,308686
landmark6	178,0052	26,76302	177,768	22,7132	4,056761
landmark7	204,7682	8,402344	203,638	10,5994	2,47071
landmark8	206,3242	34,85416	203,235	33,709	3,294624
landmark9	224,9961	65,35156	222,621	62,9984	3,343421
landmark10	188,2747	53,52604	187,407	52,3888	1,430461
landmark11	175,5156	54,45963	174,32	53,6537	1,441868
landmark12	147,1966	80,91145	140,789	81,7363	6,460473
landmark13	113,5872	92,42578	106,472	93,7031	7,228943
landmark14	69,08594	100,8281	70,1263	100,632	1,05868
landmark15	76,55469	78,73307	75,6263	77,7738	1,334956
landmark16	67,52995	61,92838	68,2324	61,1554	1,044478
landmark17	42,01172	70,95313	39,2016	68,2851	3,87494
landmark18	41,70052	90,55859	38,9611	89,6102	2,898942
			Average:		3,485863

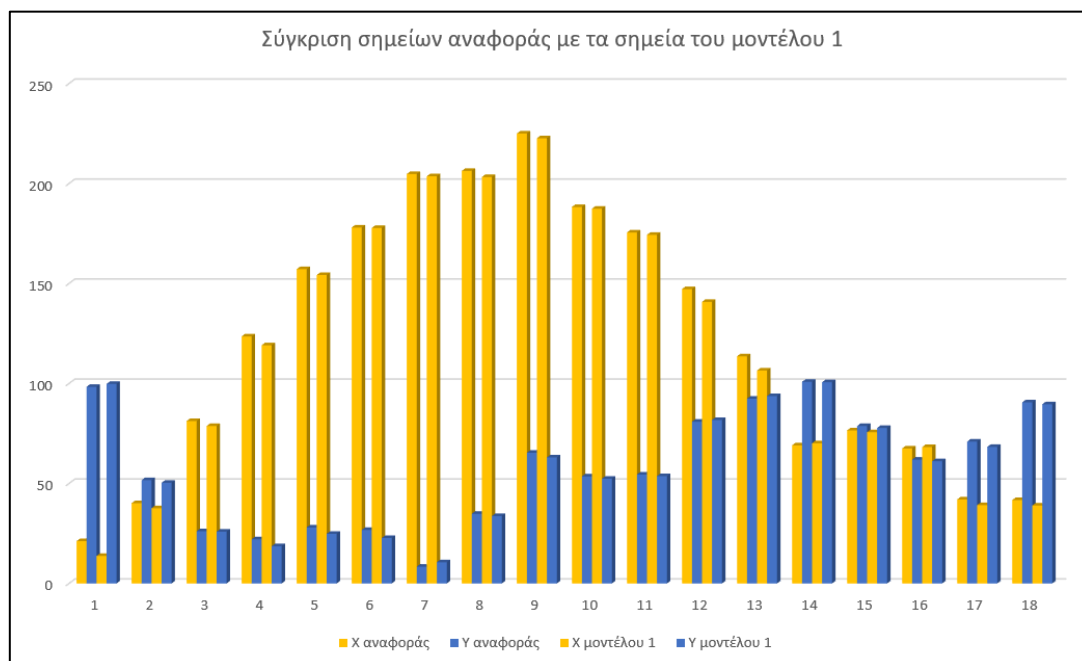
	Reference		fish2_left		Error
	X	Y	X	Y	
landmark1	21,16146	98,33854	13,7421	99,7841	0,033335
landmark2	40,14453	51,65885	37,6357	50,3691	0,015088
landmark3	81,22266	26,14063	78,7319	25,8981	0,010619
landmark4	123,5456	22,09505	119,121	18,6376	0,034451
landmark5	157,1549	28,00781	154,234	24,8403	0,029289
landmark6	178,0052	26,76302	177,768	22,7132	0,034047
landmark7	204,7682	8,402344	203,638	10,5994	0,019059
landmark8	206,3242	34,85416	203,235	33,709	0,016114
landmark9	224,9961	65,35156	222,621	62,9984	0,022131
landmark10	188,2747	53,52604	187,407	52,3888	0,010223
landmark11	175,5156	54,45963	174,32	53,6537	0,00842
landmark12	147,1966	80,91145	140,789	81,7363	0,027692
landmark13	113,5872	92,42578	106,472	93,7031	0,031647
landmark14	69,08594	100,8281	70,1263	100,632	0,004654
landmark15	76,55469	78,73307	75,6263	77,7738	0,008948
landmark16	67,52995	61,92838	68,2324	61,1554	0,00713
landmark17	42,01172	70,95313	39,2016	68,2851	0,025316
landmark18	41,70052	90,55859	38,9611	89,6102	0,01396
			Average:		0,019562

Στα παρακάτω διαγράμματα απεικονίζονται τα σημεία (landmarks) που είχαμε ως σημείο αναφοράς σε σχέση με αυτά που παρήγαγε το πρώτο μοντέλο που εκπαιδεύσαμε μαζί με το μέγεθος του σφάλματος.

## 1. Απόλυτο σφάλμα

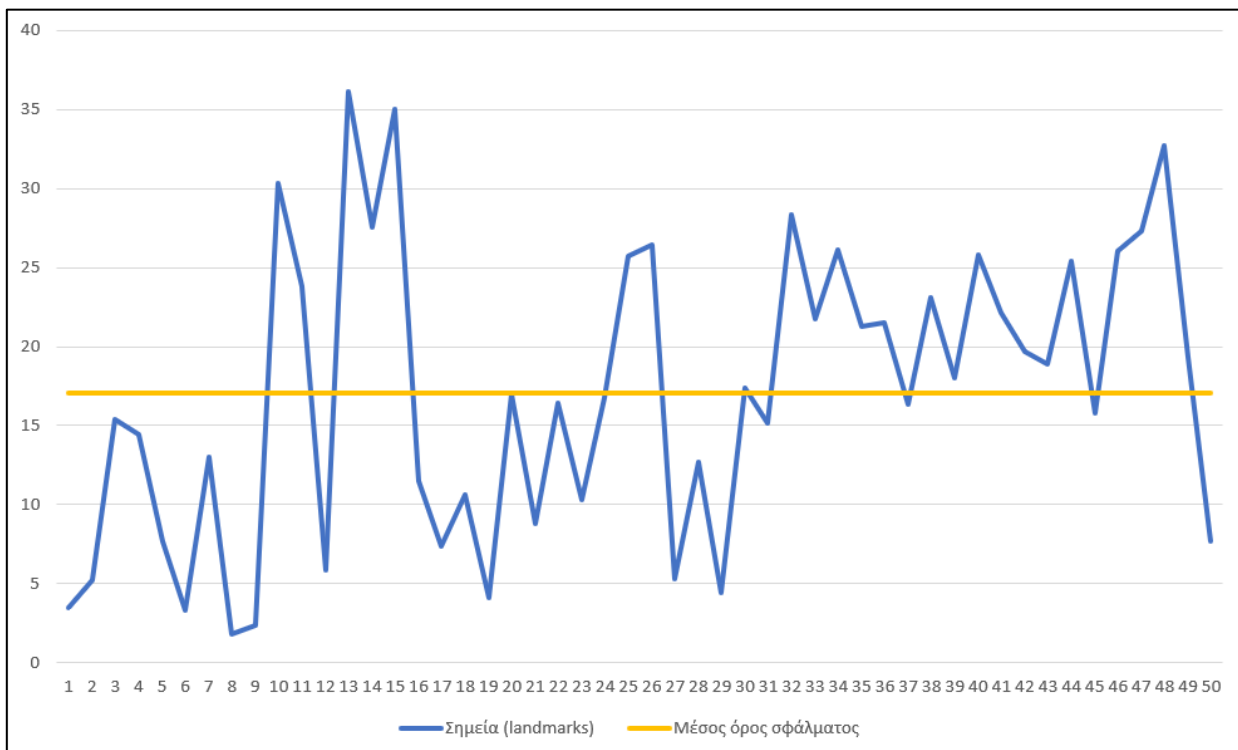


## 2. Κανονικοποιημένο σφάλμα

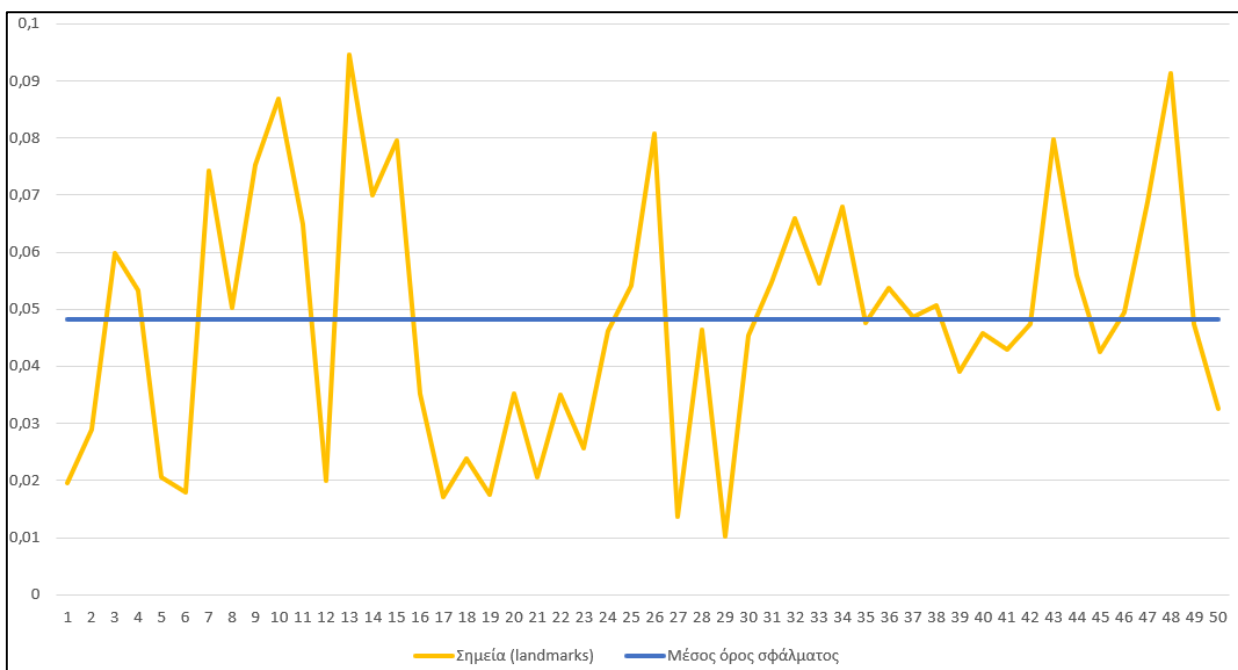


Στο παρακάτω διαγράμματα φαίνονται ο μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.



Κανονικοποιημένο σφάλμα.



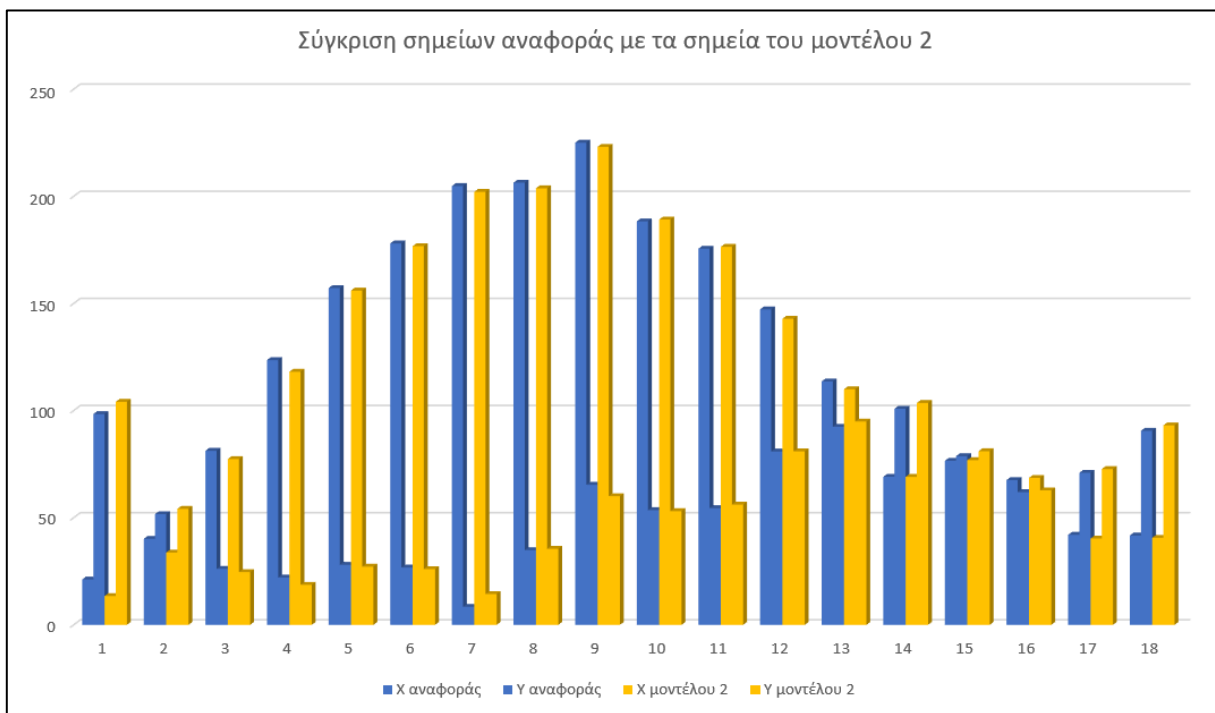


Στη συνέχεια προσπαθώντας να ανακαλύψουμε εάν η ακρίβεια και ο χρόνος εκπαίδευσης των μοντέλων θα βελτιωθούν, θα μείνουν στάσιμα ή θα μειωθούν διαφοροποιώντας κάποιες από τις παραμέτρους εκπαίδευσης όπως είναι η αλληλουχία σταδίων (cascade stages) και των δέντρων παλινδρόμησης (ERT) εκπαιδεύτηκαν ακόμα 6 μοντέλα (σύνολο 7 μαζί με το πρώτο που προαναφέρθηκε) με τα ίδια ακριβώς κριτήρια, δηλαδή με το ίδιο πλήθος 319 εικόνων, και τον έλεγχο λειτουργίας με τις ίδιες 50 εικόνες και τα 18 σημεία (landmarks) η κάθε μια από αυτές που δημιουργήθηκαν χειροκίνητα ως σημείο αναφοράς.

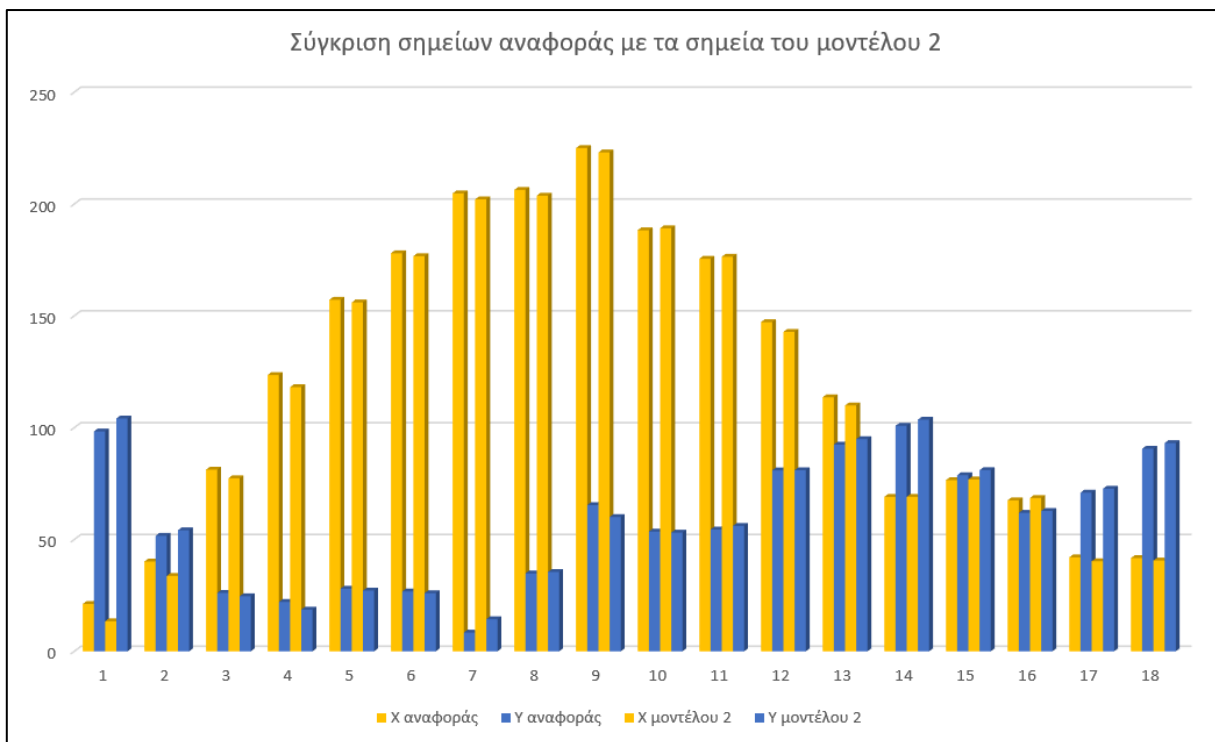
Πιο αναλυτικά το δεύτερο μοντέλο αποτελείται από μια αλληλουχία οκτώ (8) σταδίων (cascade stages) και πεντακοσίων (500) δέντρων παλινδρόμησης (ERT), Το τρίτο μοντέλο αποτελείται από μια αλληλουχία δέκα (10) σταδίων (cascade stages) και τετρακοσίων (400) δέντρων παλινδρόμησης (ERT), το τέταρτο μοντέλο αποτελείται από μια αλληλουχία οκτώ (8) σταδίων (cascade stages) και τετρακοσίων (400) δέντρων παλινδρόμησης (ERT), το πέμπτο μοντέλο αποτελείται από μια αλληλουχία έξι (6) σταδίων (cascade stages) και τετρακοσίων (400) δέντρων παλινδρόμησης (ERT), το έκτο μοντέλο αποτελείται από μια αλληλουχία οκτώ (8) σταδίων (cascade stages) και τριακοσίων (300) δέντρων παλινδρόμησης (ERT) και το έβδομο και τελευταίο μοντέλο αποτελείται από μια αλληλουχία έξι (6) σταδίων (cascade stages) και τριακοσίων (300) δέντρων παλινδρόμησης (ERT). Ακολουθώς παρατίθενται διαγράμματα από όλα τα προαναφερθέντα μοντέλα, απεικονίζοντας την σύγκριση των αποτελεσμάτων τους με τα σημεία αναφοράς σε μια από το σύνολο των 50 εικόνων που χρησιμοποιήθηκαν για τον έλεγχο των μοντέλων όπως επίσης και του μέσου όρου του σφάλματος από όλες τις εικόνες δοκιμής τόσο σε απόλυτη μορφή όσο και κανονικοποιημένη.

## Δεύτερο μοντέλο

### 1. Απόλυτο σφάλμα

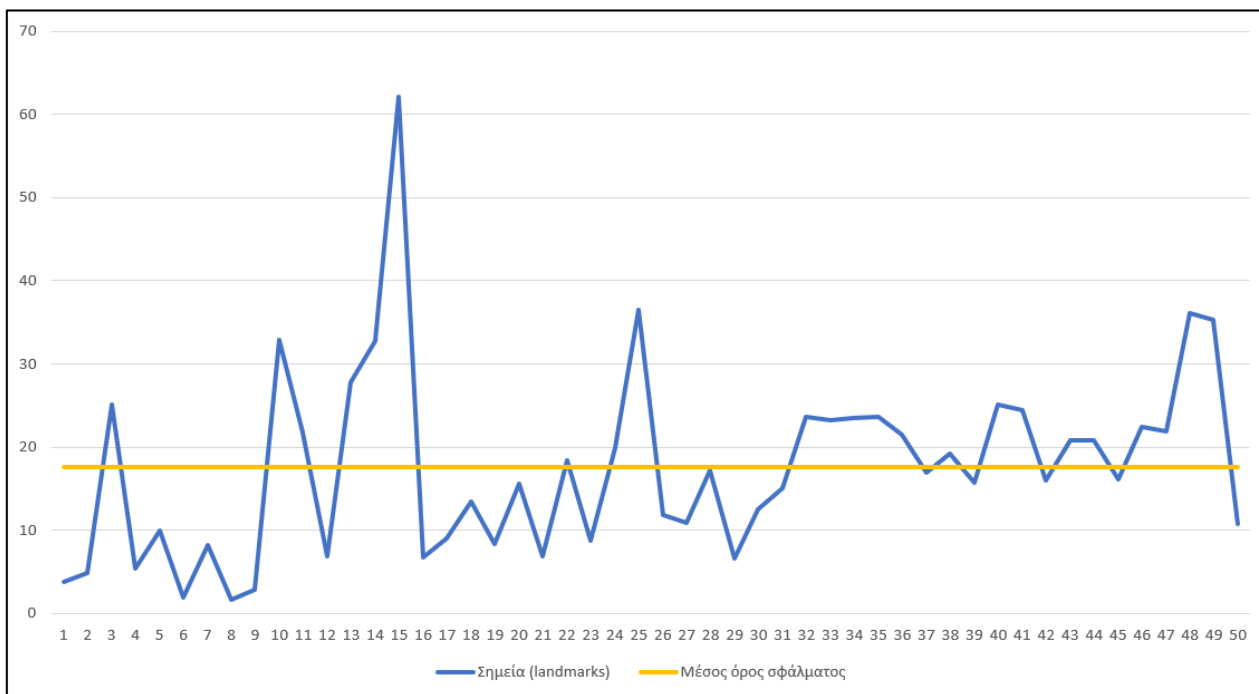


### 2. Κανονικοποιημένο σφάλμα

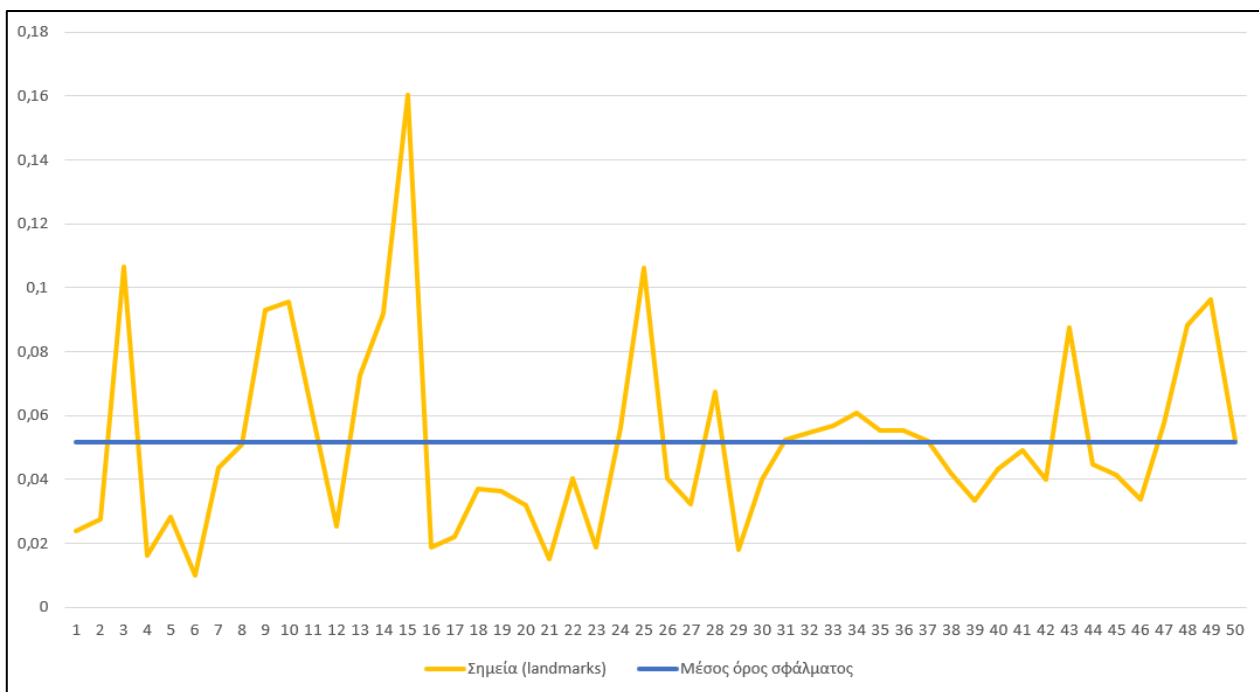


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.

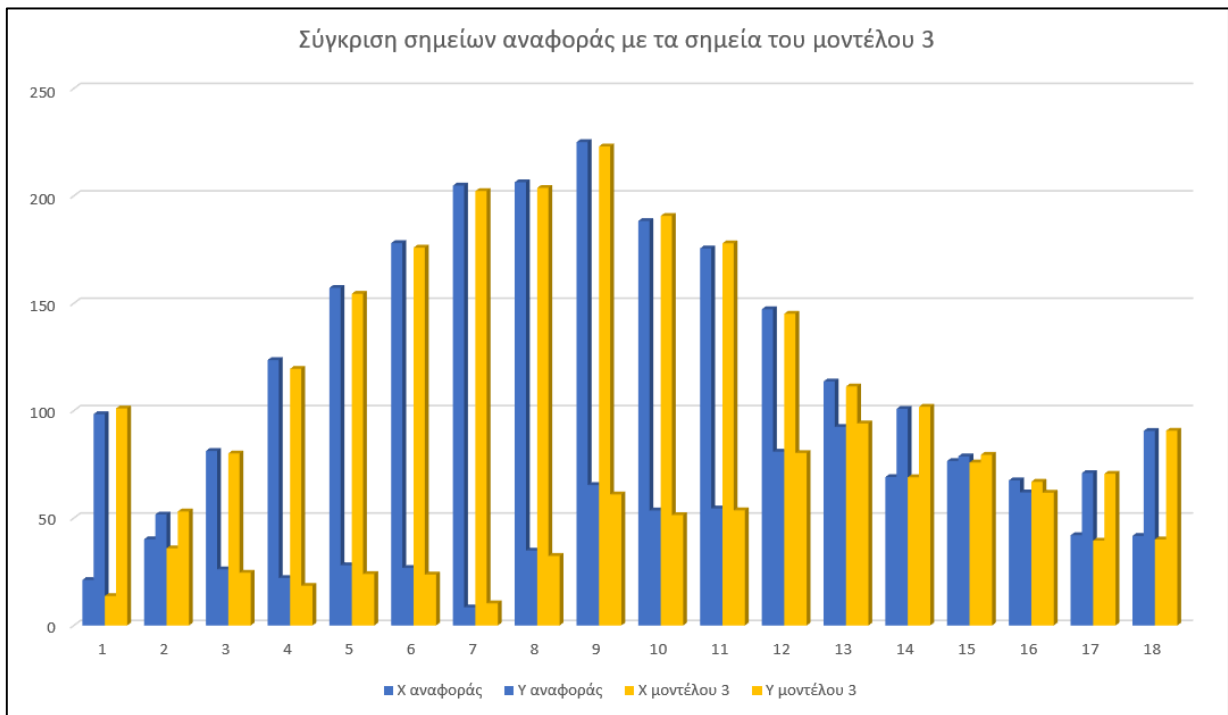


Κανονικοποιημένο σφάλμα.

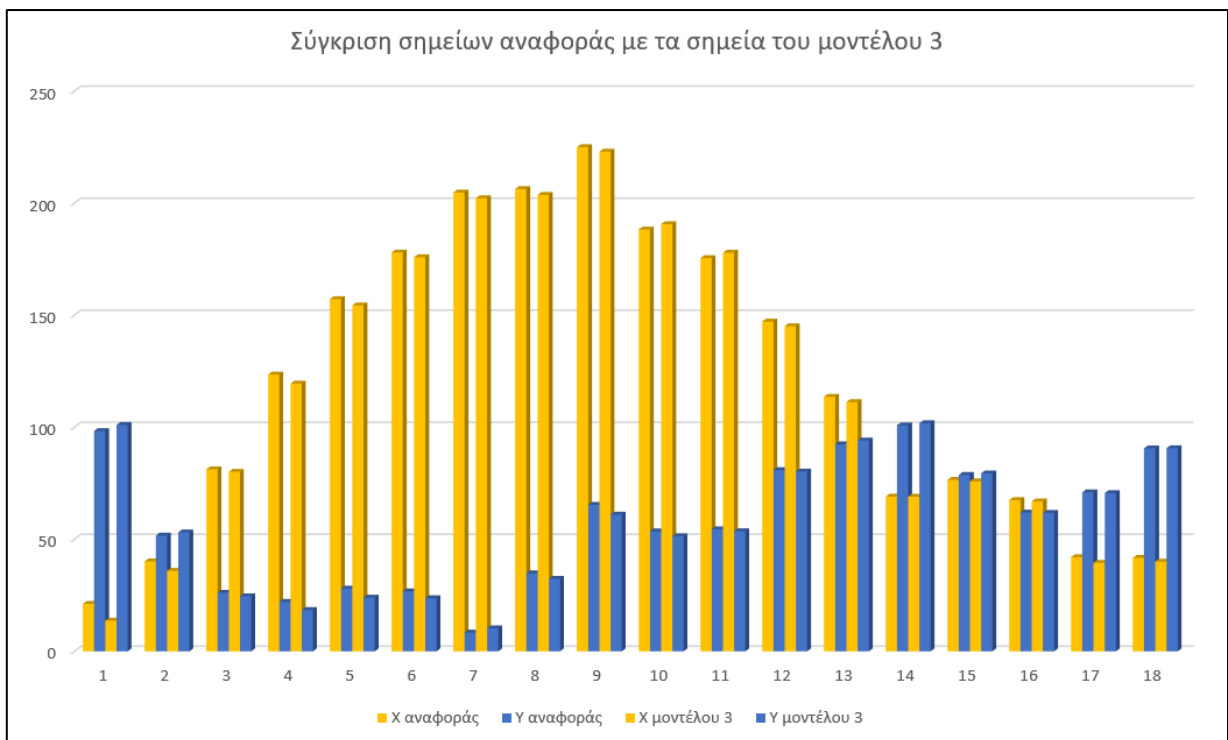


## Τρίτο μοντέλο

Απόλυτο σφάλμα

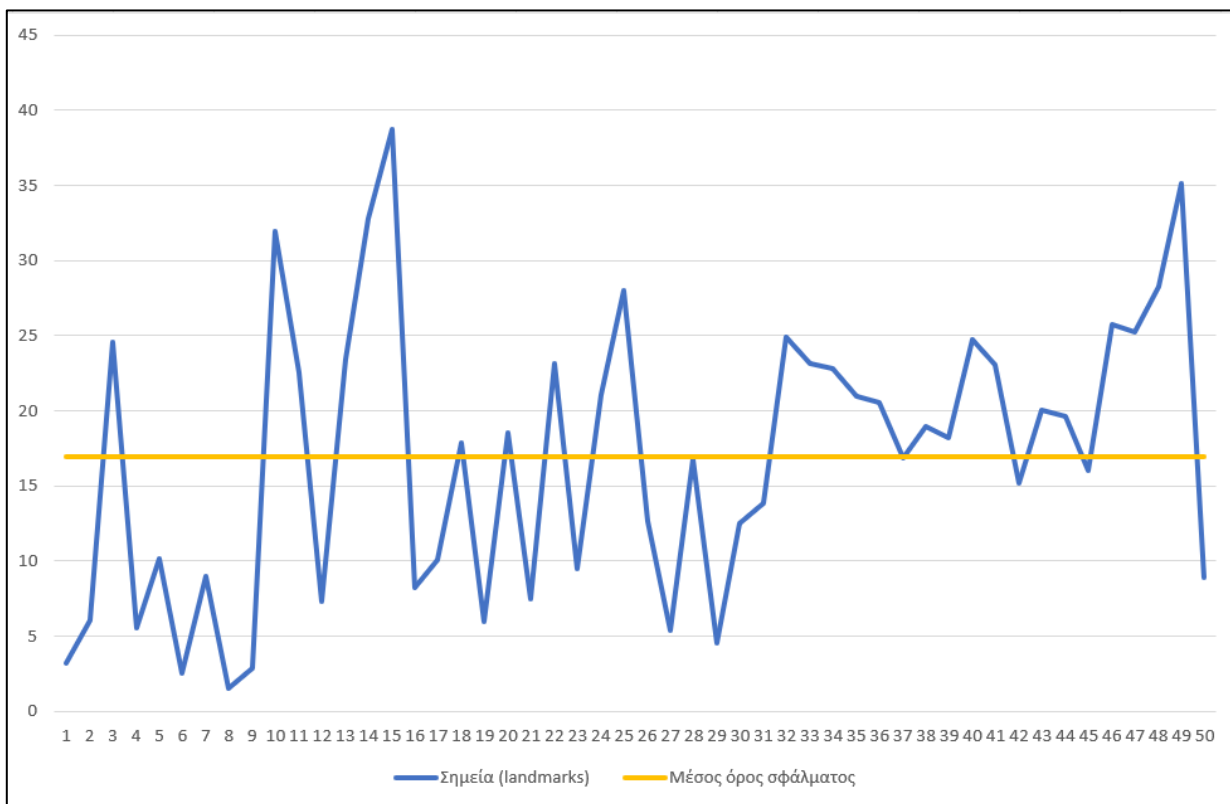


Κανονικοποιημένο σφάλμα

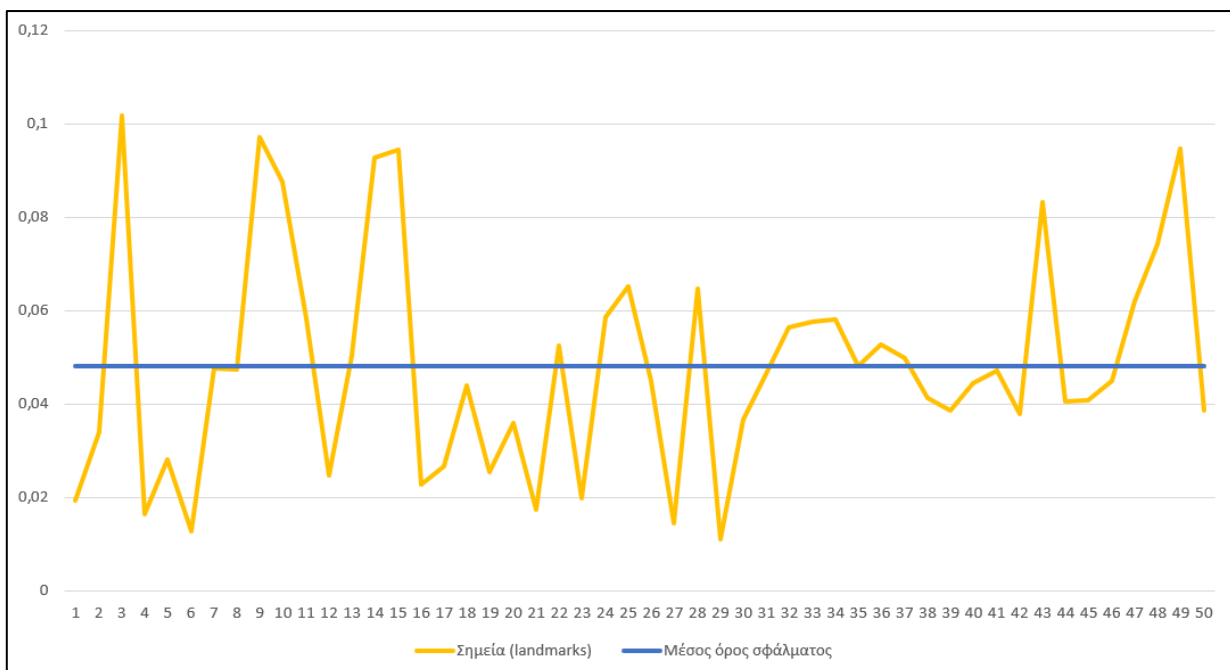


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.

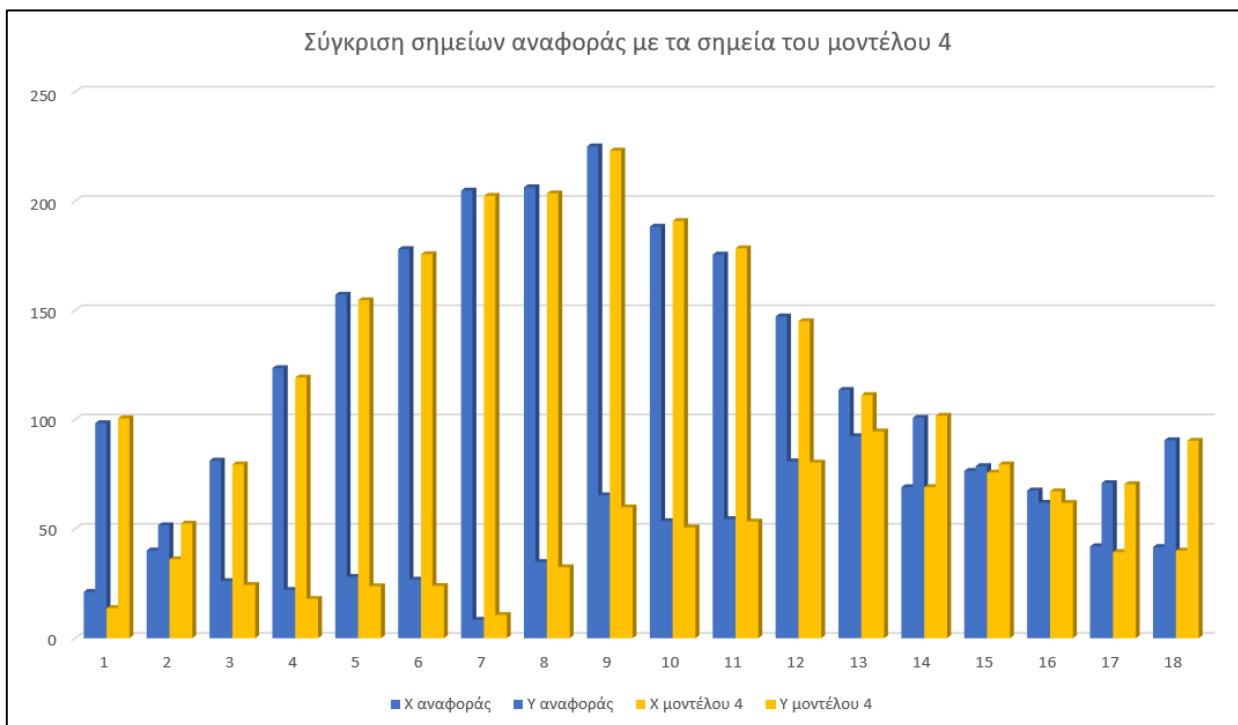


Κανονικοποιημένο σφάλμα.

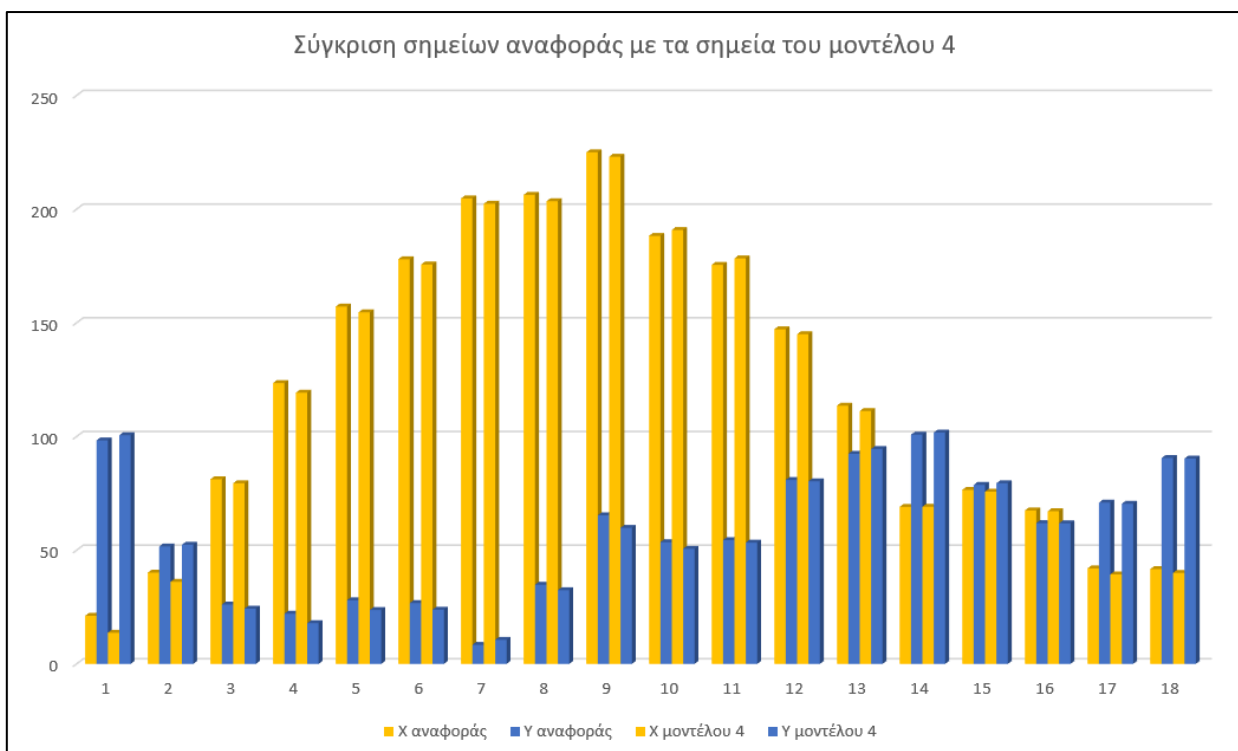


## Τέταρτο μοντέλο

### 1. Απόλυτο σφάλμα

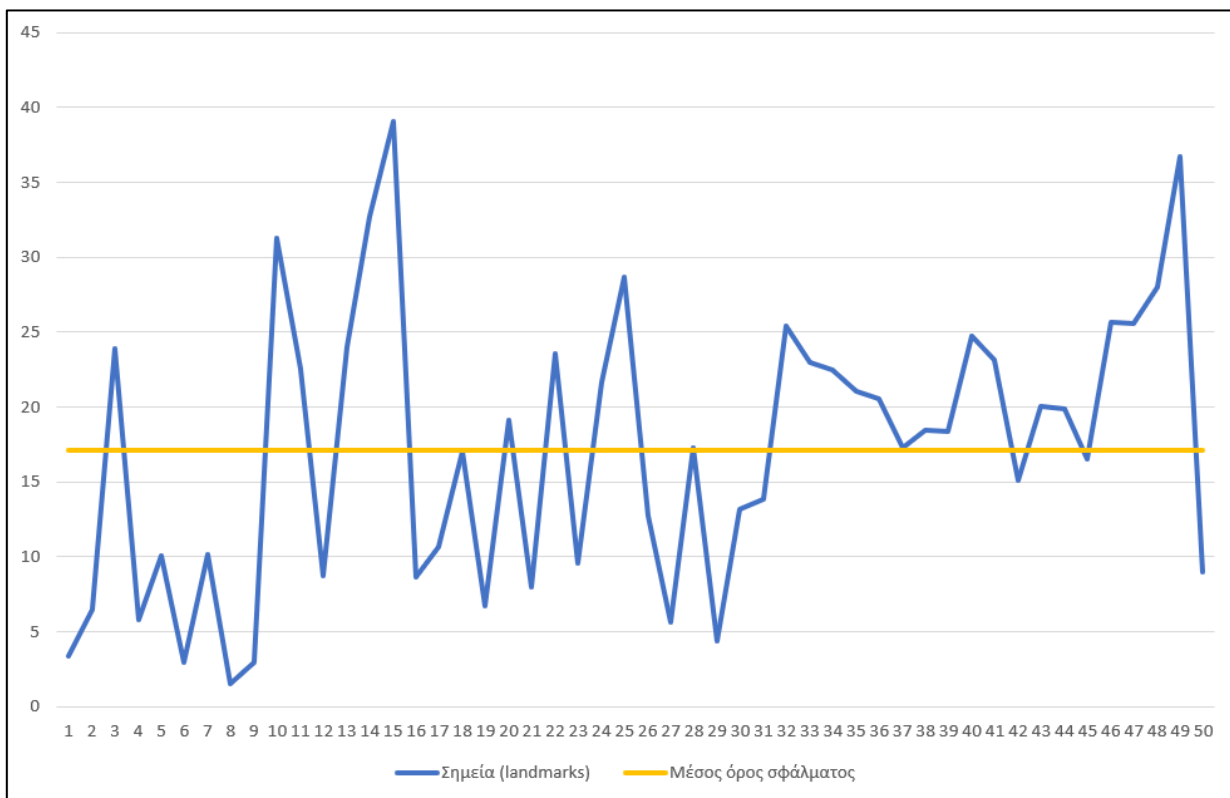


### 2. Κανονικοποιημένο σφάλμα

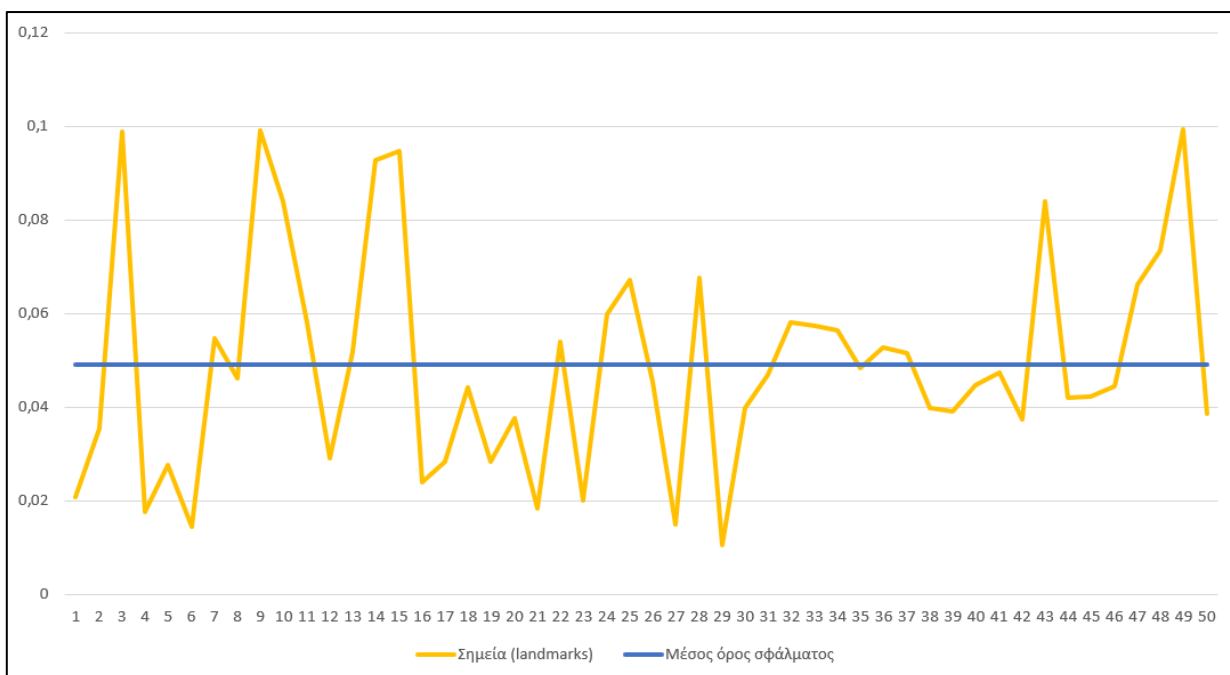


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.

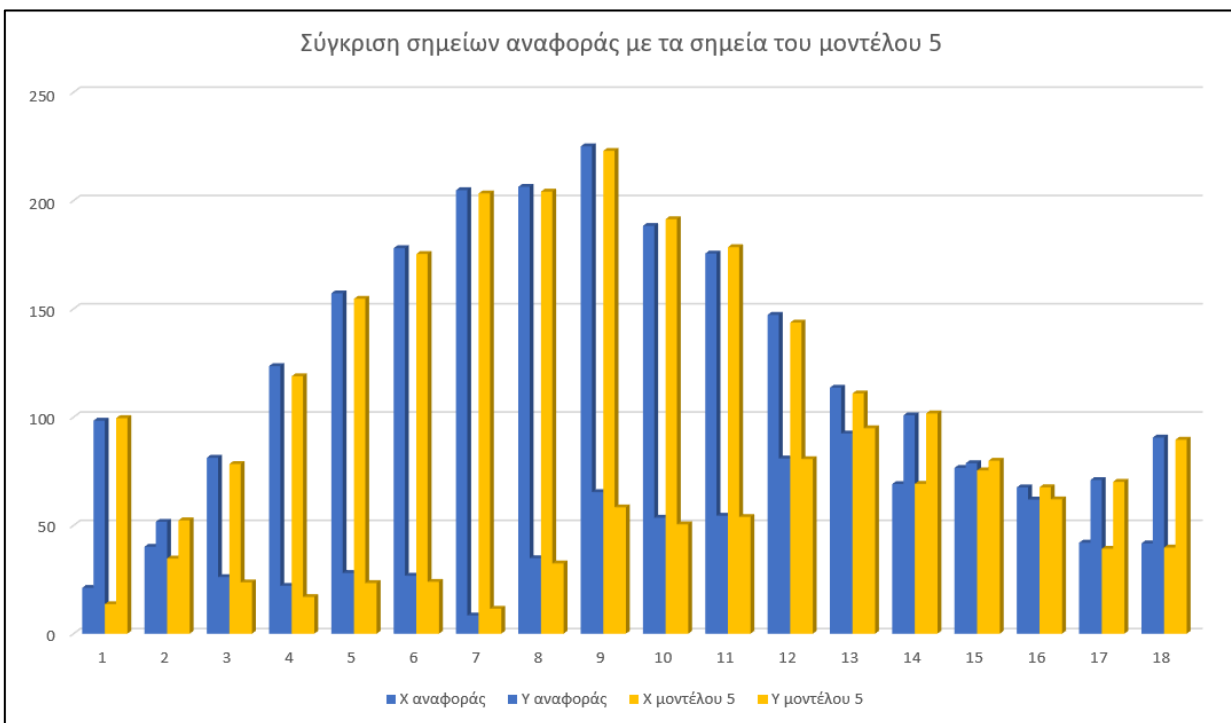


Κανονικοποιημένο σφάλμα.

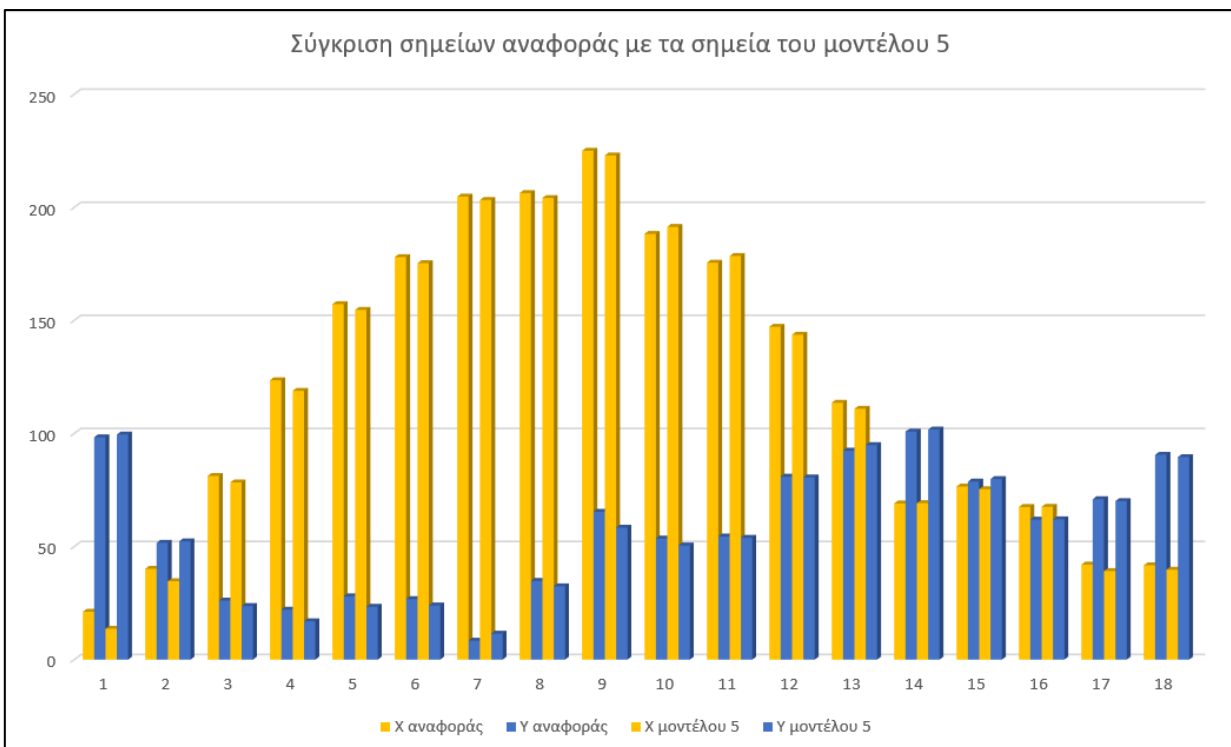


## Πέμπτο μοντέλο

### 1. Απόλυτο σφάλμα



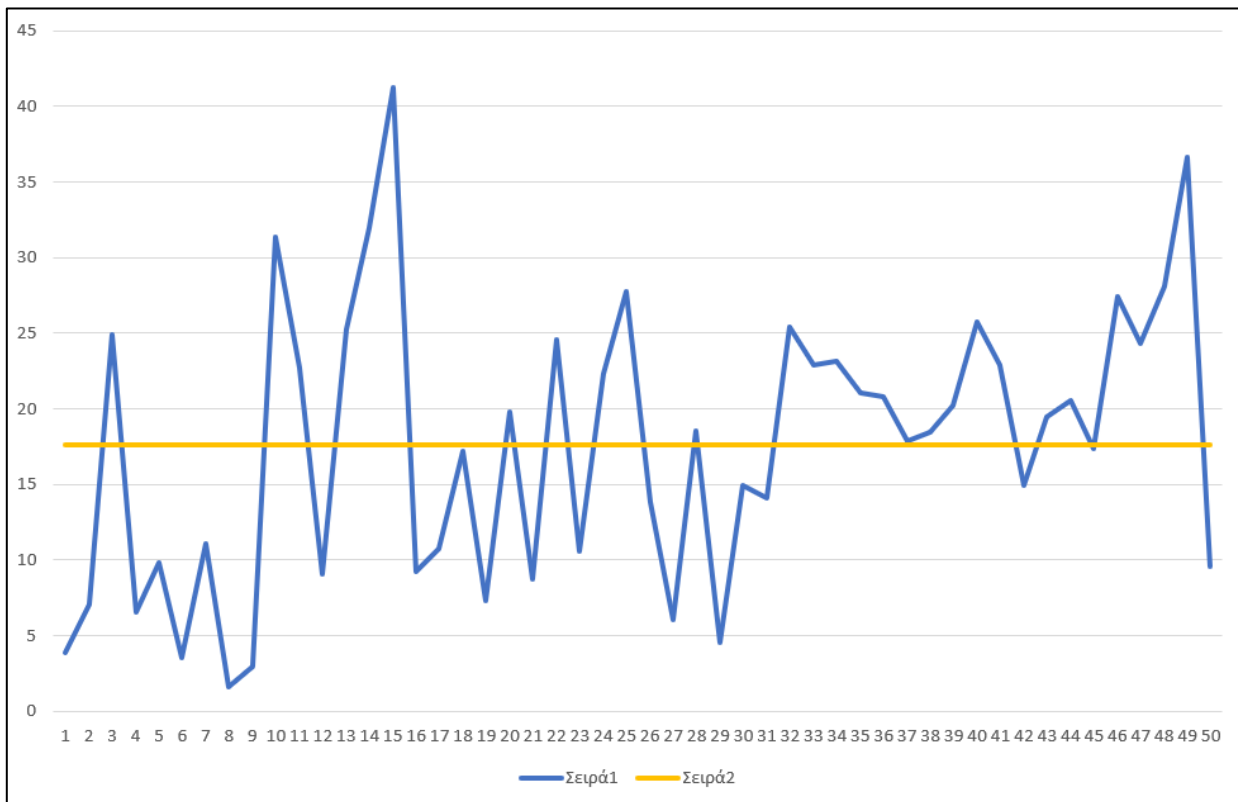
### 2. Κανονικοποιημένο σφάλμα



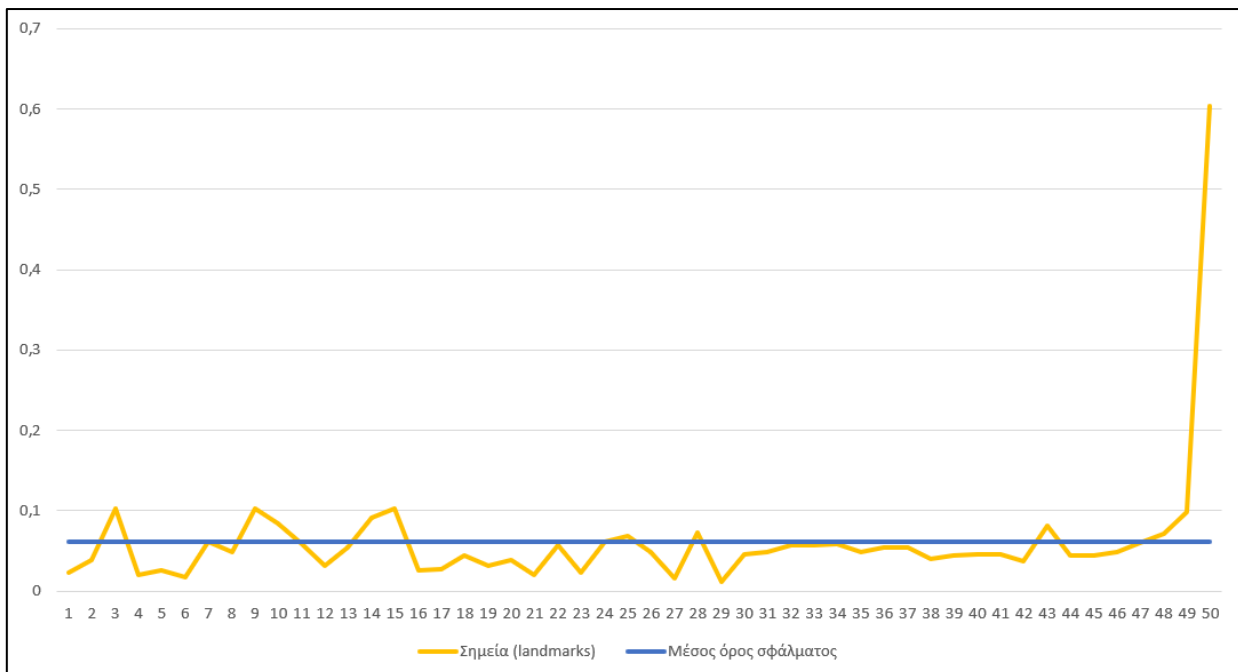


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.

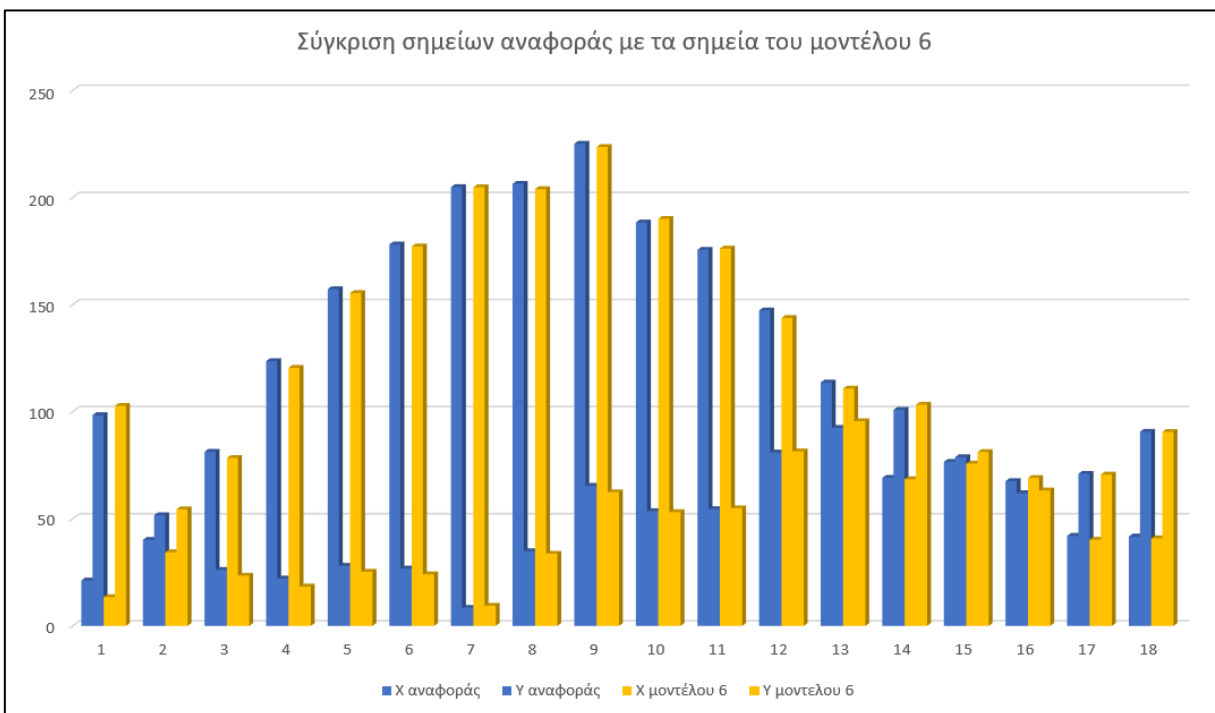


Κανονικοποιημένο σφάλμα.

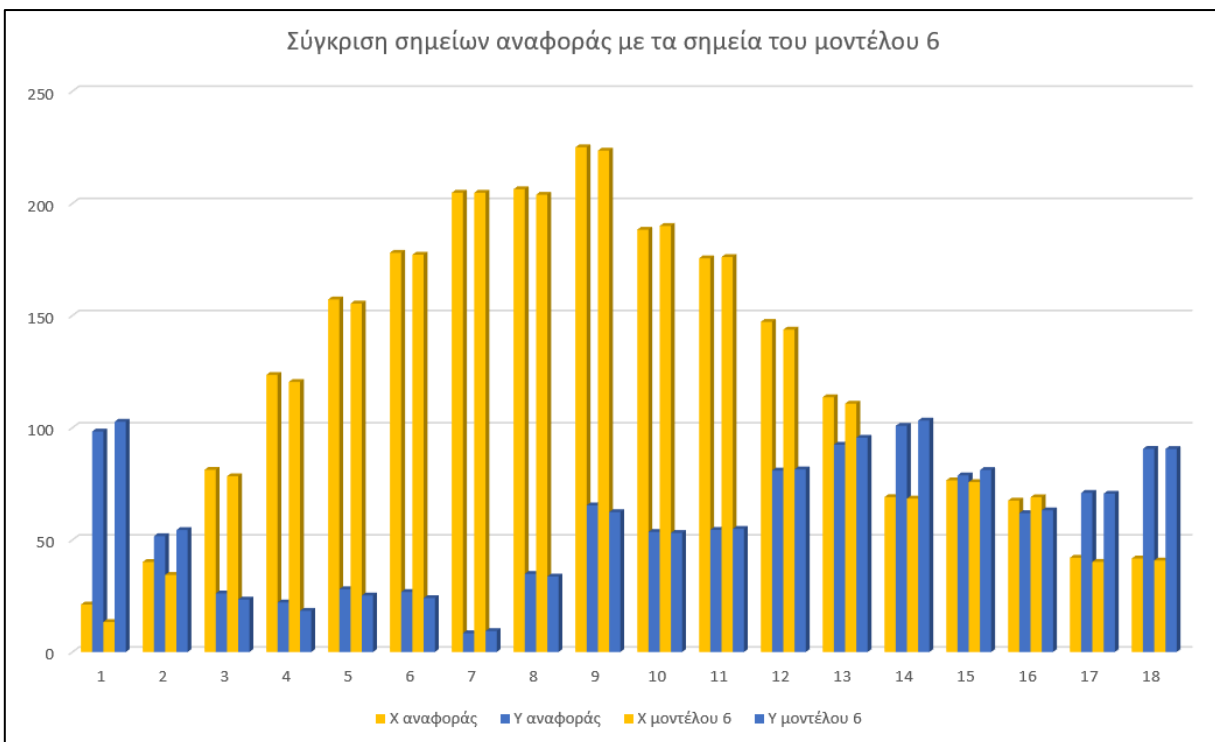


## Έκτο μοντέλο

### 1. Απόλυτο σφάλμα

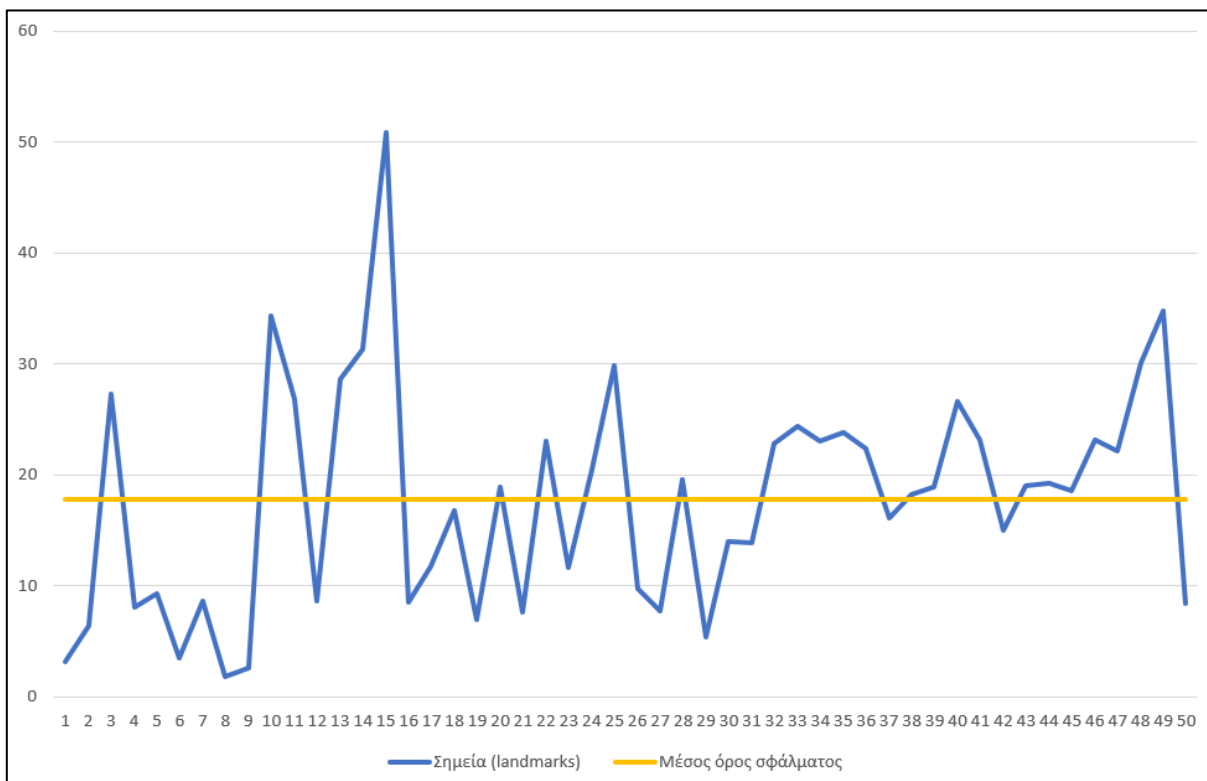


### 2. Κανονικοποιημένο σφάλμα

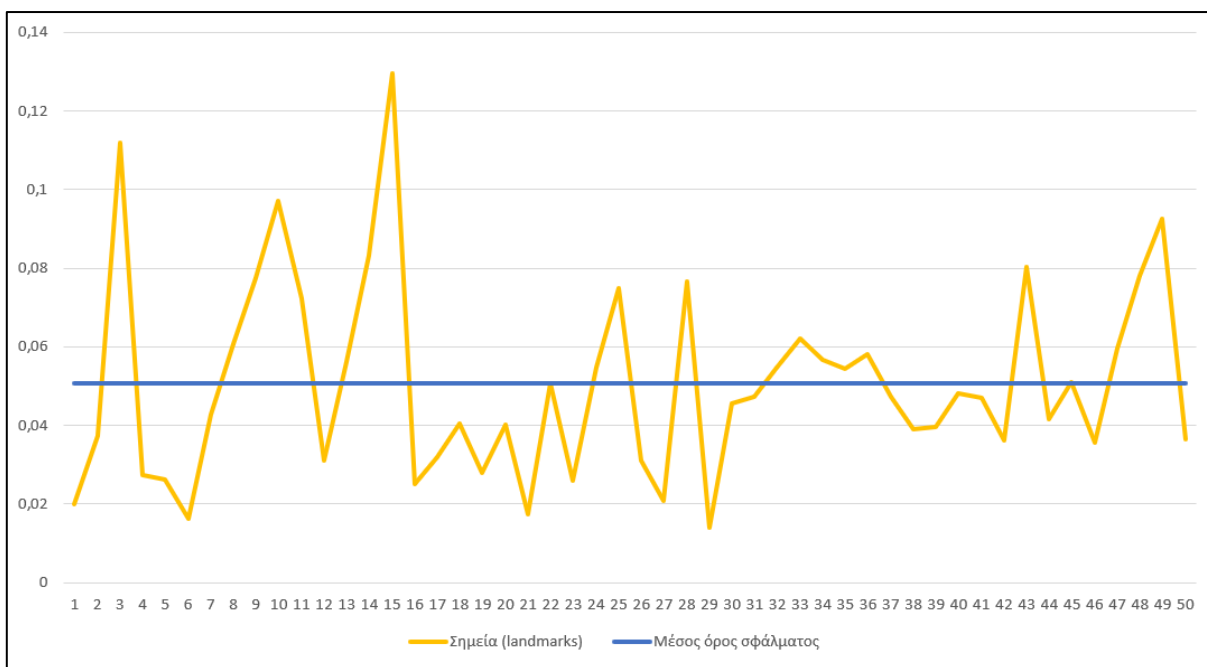


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.

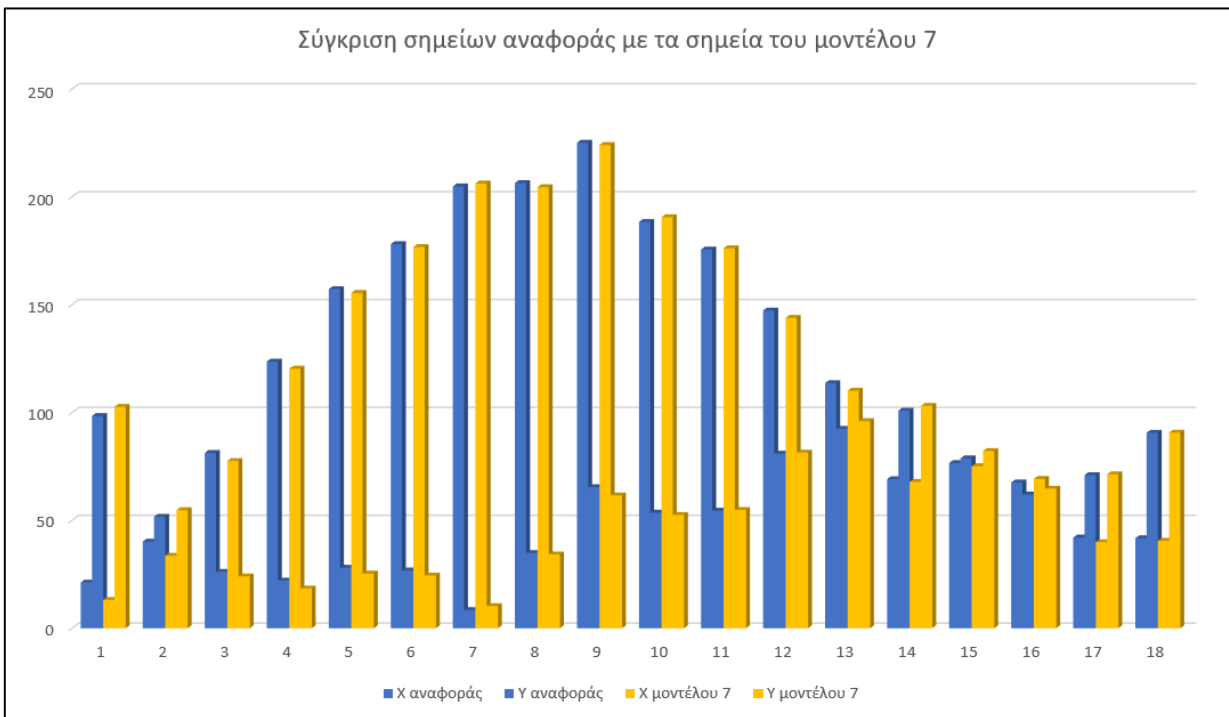


Κανονικοποιημένο σφάλμα.

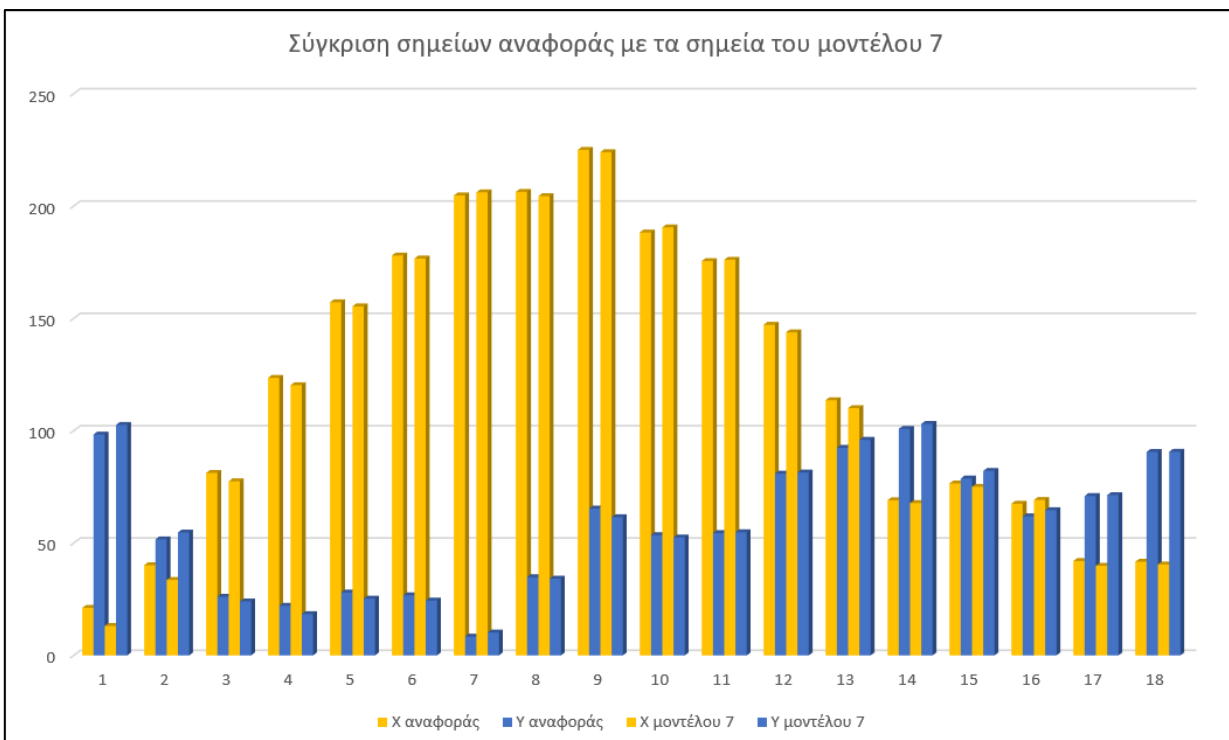


## Έβδομο μοντέλο

### 1. Απόλυτο σφάλμα

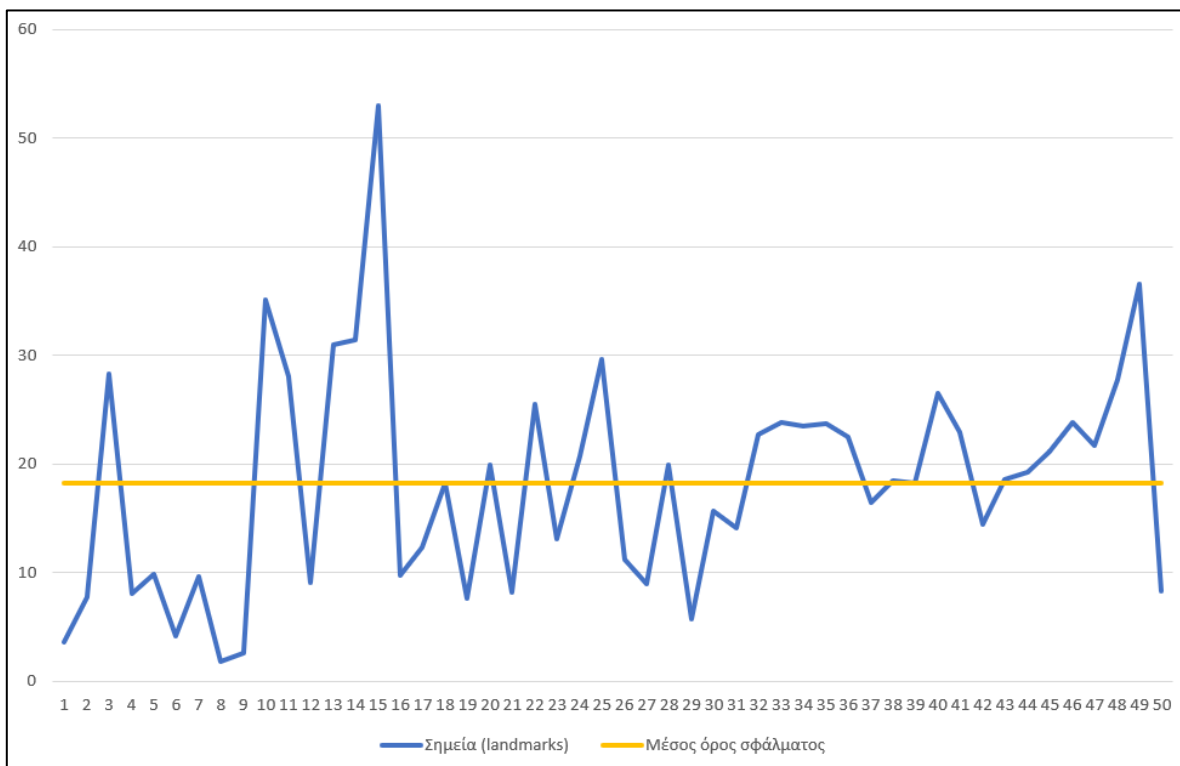


### 2. Κανονικοποιημένο σφάλμα

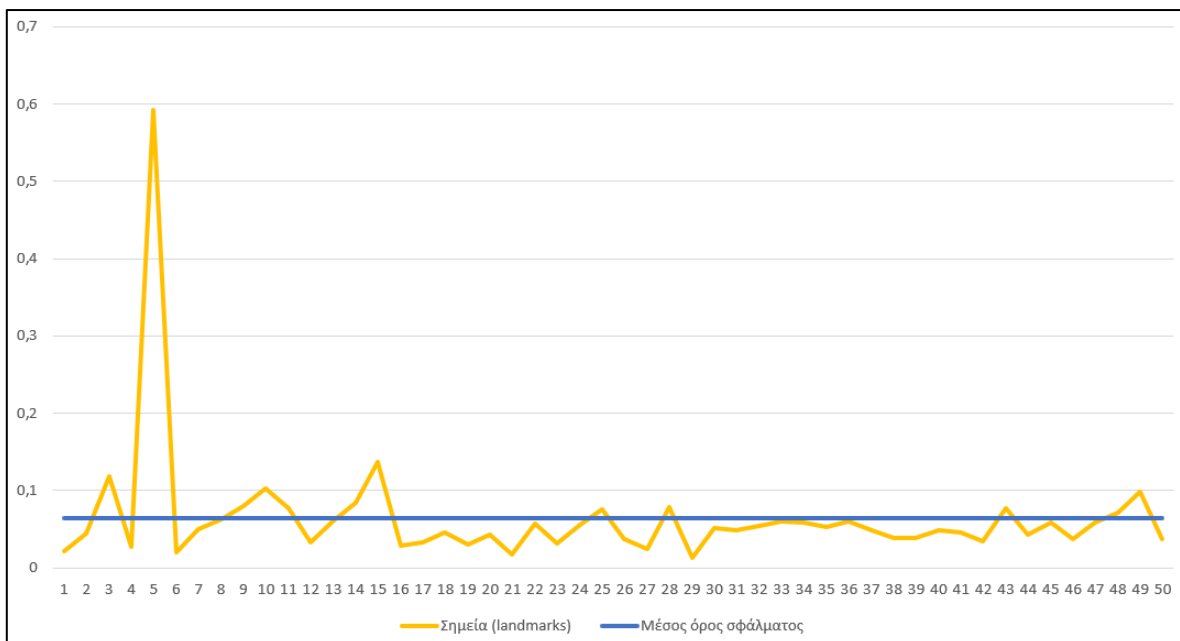


Μέσος όρος των σφαλμάτων σαν σύνολο και από τις 50 φωτογραφίες.

Απόλυτο σφάλμα.



Κανονικοποιημένο σφάλμα.



Από τα διαγράμματα και τους μέσους όρους των σφαλμάτων τελικά συμπεραίνουμε ότι στην περίπτωση του κανονικοποιημένου σφάλματος την μεγαλύτερη ακρίβεια και συνεπώς το μικρότερο ποσοστό σφάλματος το είχε το μοντέλο που εκπαιδεύτηκε με τα περισσότερα (10)-(500) στάδια αλληλουχίας (cascade stages) και δέντρα παλινδρόμησης (ERT), βέβαια αυτό δεν είναι πάντα κανόνας διότι στην περίπτωση του απόλυτου σφάλματος βλέπουμε ότι την μεγαλύτερη ακρίβεια και το μικρότερο σφάλμα την είχε το δεύτερο μοντέλο, το οποίο εκπαιδεύτηκε με 8 στάδια αλληλουχίας (cascade stages) και 500 δέντρα παλινδρόμησης (ERT). Κλείνοντας το συμπέρασμα που βγάζουμε είναι ότι δεν είναι πάντα η σωστή λύση να χρησιμοποιούμε όσο το δυνατόν μεγαλύτερο αριθμό σταδίων αλληλουχίας (cascade stages) και δέντρων παλινδρόμησης (ERT) γιατί από ένα σημείο και μετά η ακρίβεια δεν παρουσιάζει αύξηση ενώ ο χρόνος εκπαίδευσης μεγαλώνει δραματικά.

## **7. ΒΙΒΛΙΟΓΡΑΦΙΑ**

Alexandridis, T. K., Chatzigeorgiou, S., Loukis, E. N., & Manitsaris, A. (2019). Underwater fish detection using computer vision techniques. *International Journal of Computer Vision and Image Processing*, 9(3), 28-44.

Chen, Y., Zhang, H., Cao, Z., Gao, X., Chen, Q., & Li, B. (2022). A Review of Underwater Fish Observing Systems: Techniques, Applications and Future Perspectives. *Journal of Ocean University of China*, 21(3), 569-584.

Huang, M., Gao, L., Song, H., & Ji, Y. (2021). An Underwater Fish Species Recognition System Based on Deep Learning with Data Augmentation. *Journal of Marine Science and Engineering*, 9(7), 748.

Morais, R. A., Castro, J. J., Costa, M. J., & Cabral, H. N. (2013). Quantitative underwater visual census of fish populations: A review of methods and biases. *Fisheries Research*, 140, 17-26.

Smith, A. N., Mariani, S., & Bozzano, A. (2022). Advancements in Underwater Acoustic Monitoring of Fish Populations. *Sensors*, 22(6), 3381.

Sun, Z., Li, W., Zhang, Y., & Hu, J. (2020). An Underwater Fish Recognition Approach Based on Visual Saliency and Convolutional Neural Network. *Journal of Marine Science and Engineering*, 8(2), 79.