



Πανεπιστήμιο Πειραιώς
University of Piraeus

**Εφαρμογή τεχνικών μηχανικής
μάθησης για την ανάλυση
τηλεδιασκέψεων σε πραγματικό χρόνο**

Παπαδάκης Μιχαήλ

Υπεύθυνος Καθηγητής: Κυριαζής Δημοσθένης

Συνυπεύθυνος Καθηγητής: Μενύχτας Ανδρέας

Διπλωματική Εργασία, Προηγμένα Πληροφοριακά Συστήματα

Τμήμα: Ψηφιακά Συστήματα

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

Αθήνα, Ιούλιος 2020

Περίληψη

Στην παρούσα διπλωματική εξετάζεται η ιδέα και η υλοποίηση μια εφαρμογής για ζωντανές τηλεδιασκέψεις, μέσω του περιηγητή ιστού, με ζωντανό βίντεο και ήχο. Αυτές έγιναν εφικτές μέσω του πρωτοποριακού πρωτοκόλλου, WebRTC, που διεύρυνε τους ορίζοντες των κατασκευαστών περιηγητών ιστού, εφόσον δεν υπήρχε μέχρι πρότινος. Επιπλέον, τα τελευταία χρόνια έχει αυξηθεί το ενδιαφέρον για τις εφαρμογές μηχανικής μάθησης και έχουν γίνει άλματα όσον αφορά την τεχνολογία και τις εφαρμογές της. Με την εξέλιξή τους, νέες ιδέες και υλοποιήσεις προτάσσονται για να βρεθεί τι είναι εφικτό και ποια είναι τα όρια τους. Στην παρούσα διπλωματική εργασία, μελετήθηκε και αναπτύχθηκε η δημιουργία Συνελικτικών Νευρωνικών Δικτύων, η χρησιμοποίηση έτοιμων προ-εκπαιδευμένων μοντέλων και η εφαρμογή τους πάνω στον κεντρικό πυρήνα, τις βιντεοδιασκέψεις. Σκοπός της μηχανικής μάθησης είναι να δείξει στον συμμετέχοντα των βιντεοδιασκέψεων τα συναισθήματα των συμμετεχόντων σε ζωντανό χρόνο. Επειδή η εφαρμογή έχει σχεδιαστεί ώστε να εκτελείται σε περιηγητές ιστού, οι τεχνολογίες και ο τρόπος υλοποίησης αυτών αποτελεί μείζων ζήτημα στην επίτευξη ελκυστικού αποτελέσματος. Για αυτό τον σκοπό μέσα από την έρευνα που έγινε, χρησιμοποιήθηκαν εργαλεία όπως το OpenVidu, για την διαχείριση των χρηστών και των βιντεοδιασκέψεων, OpenCV για την αναγνώριση προσώπων στα βίντεο και TensorFlow, Caffe για την αναγνώριση συναισθημάτων. Η υλοποίηση έγινε κυρίως σε γλώσσα προγραμματισμού Javascript, με την χρήση της βιβλιοθήκης React, την γλώσσα προγραμματισμού Python για την εκπαίδευση των μοντέλων μηχανικής μάθησης και την ανάλυση τους σε ζωντανό χρόνο. Επιπλέον, έγιναν πειράματα σε διαφορετικές συσκευές για την εύρεση της διεκπεραιωτικότητας και τις ανάγκες της εφαρμογής, καθώς και διαφορετικές τοπολογίες και τρόπους λειτουργίας, για την ενσωμάτωσή της σε μελλοντικά σενάρια, όπως σε καταναμημένα συστήματα και νεφοϋπολογιστική.

Λέξεις Κλειδιά – Μηχανική Μάθηση, Ζωντανή βιντεομετάδοση, Ανάλυση συναισθημάτων, OpenCV, TensorFlow, Caffe, Javascript, MQTT, Python

Abstract

The present master thesis examines the idea and implementation of an application for live teleconferencing, through the web browser, with live video and audio. These were made possible through the groundbreaking WebRTC protocol which broadened the horizons of web browsers' developers, since there was no analogous before. In addition, over the last few years the interest in machine learning applications has increased and leaps ahead have been made in terms of the technology and its applications. As it evolves, new ideas and implementations are proposed, to find out its feasibility and its limits. In the present thesis, were studied and developed, the creation of Convolutional Neural Networks as well as the use of ready, pre-trained models on the video conferencing. The purpose of machine learning is to show the video conference participant, the emotions of the participants in live time. Because the application is designed to run on web browsers, technologies and implementation strategy, were crucial for an attractive result. To this end, from the research done, specific tools were used, such as OpenVidu to manage users and video conferencing, OpenCV to recognize faces in videos, and Tensorflow, Caffe to recognize emotions. The implementation was done mainly in Javascript programming language, using the React library, Python programming language for the training of machine learning of models and their analysis in real time. In addition, experiments were performed on different devices to find the feasibility and needs of the application, as well as different topologies and infrastructure, to integrate it into future scenarios, such as distributed systems and cloud computing.

Keywords – Machine Learning, Live Video Streaming, Emotion Detection, OpenCV, TensorFlow, Caffe, Javascript, MQTT, Python

Περιεχόμενα

1	Εισαγωγή	1
2	Επιστημονικό και Τεχνολογικό Υπόβαθρο	3
3	Επισκόπηση WebRTC και Μηχανικής Μάθησης	5
3.1	Αρχές του WebRTC	5
3.2	Μηχανική Μάθηση	6
3.3	Αναγνώριση συναισθημάτων σε πραγματικό χρόνο	7
3.4	Λειτουργικότητα WebRTC σε προγράμματα περιηγητή	7
3.5	Ο πυρήνας του WebRTC χωρίζεται σε τρία στοιχεία	8
3.6	Το WebRTC χωρίζεται σε τρία διαφορετικά επίπεδα API	8
3.7	Βασική επικοινωνία μέσω WebRTC	9
3.8	Επικοινωνία Peer-To-Peer	10
3.9	Σηματοδοσία, περίοδος σύνδεσης και πρωτόκολλα	11
3.9.1	NAT	13
3.9.2	ICE	14
3.9.3	STUN	14
3.9.4	TURN	14
3.9.4.1	Πότε χρειάζεται STUN	14
3.10	WebRTC Javascript API	15
3.11	Kurento Framework	15
3.12	Kurento Media Server	16
3.13	Αρχές σχεδιασμού του Kurento	17
3.13.1	Ξεχωριστά στάδια πολυμέσων και σηματοδότησης	17
3.13.2	Κατανομή πολυμέσων και υπηρεσιών εφαρμογών	17
3.13.3	Κατάλληλο για το σύννεφο	18
3.13.4	Αγωγοί Πολυμέσων	18
3.13.5	Ανάπτυξη εφαρμογής	18
3.13.6	Δυνατότητα επικοινωνίας από άκρο σε άκρο	18
3.13.7	Πλήρως επεξεργάσιμα ρεύματα πολυμέσων	18
3.13.8	Υψηλό επίπεδο προσαρμογής πολυμέσων	19
3.14	Χαρακτηριστικά του Kurento	19
3.14.1	Kurento API	19
3.14.2	Kurento API client	20
3.14.3	Τμήματα του Kurento	21
3.14.4	RTP Streaming	22
3.15	Το Kurento σε διαδικτυακή εφαρμογή	22
3.15.1	Επίπεδο παρουσίασης (πλευρά πελάτη)	23
3.15.2	Λογική εφαρμογών (πλευρά εξυπηρετητή)	23
3.15.3	Επίπεδο υπηρεσιών (διακομιστής ή διαδίκτυο)	23
3.16	Αρχιτεκτονική Client-side και Server-side	23
3.17	Αρχιτεκτονική εφαρμογών	24
3.17.1	Στάδιο σηματοδότησης (Signaling Plane)	24
3.17.2	Στάδιο πολυμέσων (Media Plane)	24
3.18	Επικοινωνία πελάτη, διακομιστή και Kurento	26
3.18.1	Εφαρμογή πελάτη	26

3.18.2	Application Server	26
3.18.3	Kurento Media Server	26
3.19	Φάση διαπραγμάτευσης των πολυμέσων (σηματοδότηση)	27
3.20	Φάση ανταλλαγής πολυμέσων	28
3.21	Εφαρμογές πραγματικού χρόνου	28
3.22	Αγωγοί πολυμέσων	29
3.23	OpenVidu	29
3.23.1	Αρχιτεκτονική	30
3.23.2	Πώς λειτουργεί το OpenVidu	30
3.23.3	Επιπλέον λειτουργίες του OpenVidu στην πλατφόρμα	32
3.24	Μηχανική Μάθηση	34
3.24.1	Αρχιτεκτονική ενός Συνελικτικού νευρωνικού δικτύου	34
3.24.2	Συνελικτική Λειτουργία	35
3.24.3	Στατιστικά των Convents και συνέλιξη	35
3.24.4	Λειτουργία μέγιστης συγκέντρωσης (Max-pooling operation)	39
3.24.5	Πολλαπλά συνελικτικά δίκτυα με Keras	40
3.24.6	Ορισμός του μοντέλου	41
3.24.7	Ακολουθιακό Μοντέλο	41
3.24.8	Συναρτησιακό Μοντέλο	41
3.24.9	Μεταγλώττιση του μοντέλου	41
3.24.10	Τοποθέτηση κατάλληλου πλαισίου	42
3.24.11	Δημιουργία πρόβλεψης	42
3.25	Confusion Matrix (Πίνακας Σύγχυσης)	42
3.25.1	Πολλαπλά συνελικτικά δίκτυα με OpenVino	43
4	Τεχνολογίες	44
4.1	Βιβλιοθήκες μηχανικής μάθησης	44
4.1.1	TensorFlow	44
4.1.2	Caffe	46
4.1.3	OpenVINO	46
4.1.4	MobileNet	46
4.2	Βιβλιοθήκες μηχανικής όρασης - OpenCV	47
4.3	Τεχνολογίες ανάπτυξης λογισμικού	48
4.3.1	Ionic	48
4.3.2	React	48
4.4	Πρωτόκολλα και Βιβλιοθήκες δικτύου	50
4.4.1	Πρωτόκολλο MQTT	50
4.4.2	Σύγκριση MQTT με HTTP	50
4.4.3	Τρόπος Λειτουργίας MQTT	51
4.4.4	Mosquitto Broker	52
4.4.5	Paho Βιβλιοθήκη	52
5	Λειτουργικές Απαιτήσεις και περιπτώσεις χρήσης	53
5.1	Λειτουργικές απαιτήσεις συστήματος	53
5.2	Διαγράμματα ροής	54
6	Αρχιτεκτονική Συστήματος	60
6.1	Επισκόπηση της διαδικασίας	61
6.1.1	Ubuntu εξυπηρετητής	61

6.1.2	Εφαρμογή Πελάτη	61
6.1.3	Ubuntu εξυπηρετητής με Intel επεξεργαστή	62
6.2	Εκκίνηση/Ακολουθία μια βιντεοδιάλεξης	62
6.3	Δομή Αρχείων	62
7	Λειτουργικότητα	65
7.1	Αρχική Σελίδα	65
7.2	Κεντρική Σελίδα	66
7.3	Περιγραφή αλγορίθμου Κεντρικής Σελίδας	67
7.3.1	Πρώτο βήμα - Βιβλιοθήκες και μοντέλα μηχανικής μάθησης	67
7.3.2	Δεύτερο βήμα - Αρχικοποιήσεις	68
7.3.3	Τρίτο βήμα - Σύνδεση με το σύστημα Paho και MQTT	68
7.3.4	Τέταρτο βήμα - Είναι η εκκίνηση της διαδικασίας μηχανικής μάθησης	68
7.4	Περιγραφή κώδικα στον διακομιστή Ubuntu με το MQTT	70
8	Μοντέλα Μηχανικής Μάθησης σε πραγματικό χρόνο	71
8.0.1	Πολλαπλά συνελκτικά δίκτυα με Keras	71
8.0.2	Πολλαπλά συνελκτικά δίκτυα με OpenVino	72
8.1	Δημιουργία Συνάρτησης μηχανικής μάθησης με Mobilenet	72
8.1.1	Κανονικοποίηση των δεδομένων	73
8.1.2	Ορισμός του Συνελκτικού Νευρωνικού Δίκτυο	73
8.1.3	Επίπεδο Εισαγωγής εικόνων	73
8.1.4	Επίπεδο συνέλιξης	73
8.1.5	Επίπεδο Πρόβλεψης	74
8.1.6	KNN - Ταξινομητής	74
9	Πειράματα και αποτελέσματα	75
9.1	Πείραμα με το λάπτοπ Asus VivoBook	76
9.1.1	Πείραμα με χρήση OpenVino	76
9.1.2	Πείραμα με χρήση Keras	77
9.2	Πείραμα με το smartphone Xiaomi Mi A1	77
9.2.1	Πείραμα με χρήση OpenVino	77
9.2.2	Πείραμα με χρήση Keras	78
9.3	Πείραμα με το smartphone Samsung Galaxy A40	78
9.3.1	Πείραμα με χρήση OpenVino	78
9.3.2	Πείραμα με χρήση Keras	79
10	Μελλοντικές Επεκτάσεις και μελέτες	85
	Βιβλιογραφία	86

Κατάλογος σχημάτων

3.1	Τυπική επικοινωνία μέσω WebRTC μέσω signaling server	10
3.2	Σήματα ανταλλαγής πληροφορίας για την επικοινωνία δύο χρηστών	12
3.3	Σενάρια επικοινωνίας μεταξύ διαφορετικών πελατών	16
3.4	Ενθυλάκωση λειτουργιών του WebRTC και ενσωμάτωση καινούργιων	17
3.5	Παράδειγμα του Kurento Clients σε τρία σενάρια: 1.Χρήση του Client Kurento Java σε διακομιστή εφαρμογών Java EE. 2.Χρησιμοποιώντας το πρόγραμμα-πελάτη Javascript Kurento σε ένα διακομιστή Node.js. 3.Σύνδεση πελατών Kurento (Java και Javascript) στο Kurento Media Server	20
3.6	Στοιχεία πολυμέσων του Kurento	21
3.7	Αρχιτεκτονική μονάδων Kurento. Ο διακομιστής Kurento Media Server μπορεί να επεκταθεί με ενσωματωμένες μονάδες (ανιχνευτής πλήθους, ανιχνευτής ποντικιού, χρώμα, platedetector) και επίσης με άλλες προσαρμόσιμες ενότητες.	22
3.8	Αρχιτεκτονική εφαρμογών web και πολυμέσων σε λογικά επίπεδα.	24
3.9	Εννοιολογική αναπαράσταση της αρχιτεκτονικής του Kurento σε υψηλό επίπεδο	25
3.10	Δημιουργία αγωγών επικοινωνίας	27
3.11	Φίλτρα στην επικοινωνία ανάμεσα από τους χρήστες	29
3.12	Λειτουργία αγωγού πολυμέσων	29
3.13	Εφαρμογές στο εμπρόσθιο κομμάτι και στον εξυπηρετητή	30
3.14	Λειτουργία συνελκτικού δικτύου σε μια εικόνα	34
3.15	Λειτουργία συνέλιξης	36
3.16	Αποτέλεσμα συνέλιξης: το μέγεθος του πίνακα είναι 3x3 και ο πίνακας εξόδου είναι το αποτέλεσμα της λειτουργίας μεταξύ του πίνακα της εικόνας και του φίλτρου	37
3.17	Εφαρμογή της συνέλιξης σε όλα τα pixels	38
3.18	Συνέλιξη με βηματισμό 1	39
3.19	Συνέλιξη με βηματισμό 2	39
3.20	Συγκέντρωση με βάση το μέγιστο στοιχείο	40
4.1	Τρόπος λειτουργίας React Component	49
4.2	Το μοντέλο MQTT δημοσιεύει και εγγράφει μηνύματα από και προς αισθητήρες IoT	52
5.1	Διευθυνσιοδότηση με τον διακομιστή ICE/TURN	54
5.2	Διαχείριση βίντεο-μεταδόσεων	55
5.3	Σύνδεση στο σύστημα MQTT	56
5.4	Τοπική μηχανική μάθηση	57
5.5	Απομακρυσμένη μηχανική μάθηση	58
6.1	Διάγραμμα συστήματος της αρχιτεκτονικής της πλατφόρμας και των διακομιστών	60
7.1	Αρχική σελίδα	65
7.2	Βίντεο μετάδοση μεταξύ δύο χρηστών	66
7.3	Τοπική μηχανική μάθηση	67
7.4	Κεντρικός βρόγχος μηχανικής μάθησης	69

Κατάλογος πινάκων

3.1	Πίνακας Σύγχυσης Keras	42
3.2	Πίνακας Σύγχυσης Auto-Keras	43
3.3	Λεπτομέρειες μοντέλου OpenVino	43
9.1	Πίνακας αποτελεσμάτων με λάπτοπ, OpenVino	77
9.2	Πίνακας αποτελεσμάτων με λάπτοπ, Keras	77
9.3	Πίνακας αποτελεσμάτων με smartphone Xiaomi Mi A1, OpenVino	78
9.4	Πίνακας αποτελεσμάτων με smartphone Xiaomi Mi A1, Keras	78
9.5	Πίνακας αποτελεσμάτων με smartphone Samsung Galaxy A40, OpenVino	79
9.6	Πίνακας αποτελεσμάτων με smartphone Samsung Galaxy A40, Keras	79

1 Εισαγωγή

Ένα συναίσθημα καθορίζεται από την φυσιολογική και πνευματική κατάσταση ενός ατόμου. Είναι αποτέλεσμα πολλών εξαρτημένων μεταβλητών και εκφράζεται με πολλούς τρόπους. Όμως ο κάθε άνθρωπος δεν εκφράζεται με τον ίδιο τρόπο. Τα ανθρώπινα συναισθήματα εκφράζονται κατά 60% από τις εκφράσεις του προσώπου. Εφόσον τα συναισθήματα παίζουν μεγάλο ρόλο στην επικοινωνία, είναι επόμενο να ασχοληθεί η επιστήμη με την αναγνώρισή τους. Ενώ ένας άνθρωπος μπορεί να καταλάβει τα συναισθήματα ενός άλλου ανθρώπου, με βάση την φυσιογνωμικές αλλαγές της έκφρασης του προσώπου, είναι αρκετά πιο περίπλοκη υπόθεση για τον υπολογιστή.

Έχουν γίνει πολλές έρευνες για την εξαγωγή αποτελεσμάτων πάνω σε συναισθήματα. Από αυτές ξεχωρίζουν, όσες χρησιμοποιούν Συνελικτικό Νευρωνικό Δίκτυο (Convolutional Neural Networks). Αυτά είναι τα πιο αποτελεσματικά νευρωνικά δίκτυα για δεδομένα εικόνας και χρησιμοποιούνται συχνά και για δεδομένα ήχου. Αυτό συμβαίνει κυρίως επειδή τα δεδομένα εικόνας και ήχου αποτελούνται από τεράστιο όγκο πληροφοριών, είναι συνήθως θορυβώδη και έχουν πληροφορίες σχετικές με το περιβάλλον. Τα Συνελικτικό Νευρωνικό Δίκτυο, τα οποία είναι σύνθετα δίκτυα προώθησης πληροφορίας, είναι ικανά να δέχονται μεγάλα δεδομένα εισόδου, να εξάγουν αποτελεσματικά τα επιθυμητά χαρακτηριστικά και να αποτρέπουν την υπερβολική προσαρμογή τους στα δεδομένα εκπαίδευσης. Επομένως, τα Συνελικτικό Νευρωνικό Δίκτυο επιλέχθηκαν, για τα μοντέλα τους, μετά από σύγκριση με άλλες τεχνικές βαθιάς μάθησης όπως το Support Vector Machine (SVM). Οι επιχειρήσεις που έχουν πελατειακή μορφή, τους είναι κρίσιμο να κατανοήσουν και να μετρήσουν την ικανοποίηση των πελατών τους για τα προϊόντα ή τις υπηρεσίες που προσφέρουν. Θέλοντας να κατανοήσουν τους πελάτες τους, χρησιμοποιούν δύο τρόπους σε κείμενο, είτε με την μορφή ερωτηματολογίου, είτε με την διατύπωση σχολίων. Επειδή, τα εργαλεία αυτά όμως, έχουν αποδειχθεί αρκετά δύσκολα και ο κόσμος κατά κόρον δεν τα χρησιμοποιεί, έχει τεθεί σε πειραματικό επίπεδο η μηχανική μάθηση. Αναγνωρίζοντας τα συναισθήματα των καταναλωτών, μέσω εικόνας, ήχου και συμπεριφοράς έχουν έρθει στο προσκήνιο νέες ιδέες για το πώς πρέπει να αντιμετωπίζεται η εξατομικευμένη πλέον εξυπηρέτηση των πελατών. Πιο συγκεκριμένα, έχουν εφαρμοστεί σε:

- προγράμματα αναπαραγωγής βίντεο με βάση τη διάθεση,
- ανίχνευση ανθρώπινης-μη φυσιολογικής συμπεριφοράς
- έξυπνες αλληλεπιδράσεις ανθρώπου-υπολογιστή.

Με δεδομένες τις νέες τεχνολογίες και για την εφαρμογή σε μελλοντικές, χρησιμοποιήθηκε το πρωτόκολλο WebRTC (Web Real-Time Communication), που λειτουργεί στους περιηγητές ιστού και είναι προσβάσιμο ευρέως. Με μία σύνδεση στο internet και με μία κάμερα, είναι εφικτή η δημιουργία βιντεοδιασκέψεων. Από εκεί μπορεί ο κάθε χρήστης να κάνει σε πραγματικό χρόνο, είτε τοπική ανάλυση, είτε ανάλυση σε απομακρυσμένο εξυπηρετητή, συναισθημάτων στα ζωντανά μεταδιδόμενα βίντεο των συμμετεχόντων.

2 Επιστημονικό και Τεχνολογικό Υπόβαθρο

Από τον 19ο αιώνα, ένα από τα σημαντικότερα έργα ανάλυσης έκφρασης προσώπου που έχει άμεση συσχέτιση με την τρέχουσα υπερσύγχρονη μέθοδο αναγνώρισης έκφρασης προσώπου ήταν η προσπάθεια που έγινε από τον Charles Darwin, ο οποίος το 1872, έγραψε μια διατριβή που αναγνώρισε τις γενικές αξίες της έκφρασης και τις εκφράσεις τόσο σε ανθρώπους όσο και σε ζώα. Ο Δαρβίνος ομαδοποίησε διάφορα είδη, σε όρους και σε όμοιες ομάδες. Η ταξινόμηση έχει ως εξής:

1. χαμηλό ηθικό, άγχος, θλίψη, απογοήτευση και απελπισία.
2. χαρά, υψηλό ηθικό, αγάπη, τρυφερά συναισθήματα και αφοσίωση.
3. προβληματισμός, διαλογισμός, κακή συμπεριφορά και κατσούφιασμα
4. μίσος και θυμός
5. περιφρόνηση, αηδία, ενοχή και υπερηφάνεια.
6. έκπληξη, κατάπληξη, φόβος και τρόμος.
7. αυτό-προσοχή, ντροπή, ντροπαλότητα και σεμνότητα.

Ένα άλλο σημαντικό ορόσημο στην έρευνα της αναγνώρισης των εκφράσεων του προσώπου και των ανθρώπινων συναισθημάτων ήταν έργο του Paul Ekman και των συναδέλφων το τη δεκαετία του 1970. Ο Paul Ekman και οι συνάδελφοί του, ταξινόμησαν τα βασικά συναισθήματα και η δουλειά τους είχε σημαντική αντίκτυπο στην εξέλιξη της ανάλυσης συναισθημάτων. Η ανάλυση συναισθηματικής κατάστασης ανάγεται στην ψυχολογία. Ωστόσο, η ανάλυση σημάτων και εικόνας, με μεθόδους επεξεργασίας και αναγνώρισης μοτίβων, μπορούν να διαδραματίσουν θεμελιώδη ρόλο στον υπολογισμό των συναισθημάτων. Πρώτα, η συναισθηματική κατάσταση ενός ατόμου μπορεί να ανιχνευθεί από την έκφραση του προσώπου, την ομιλία και τις χειρονομίες του σώματος με συστήματα εικόνων. Δεύτερον, τα χαρακτηριστικά μπορούν να εξαχθούν από αυτές τις ηχογραφήσεις βάση μεθόδων επεξεργασίας σήματος και εικόνας. Τέλος, εφαρμόζονται προηγμένες μέθοδοι αναγνώρισης προτύπων για την αναγνώριση των συναισθηματικών καταστάσεων.

Η πρώτη μελέτη της αυτόματης αναγνώρισης έκφρασης προσώπου πραγματοποιήθηκε το 1978 από τον Suwa κ.α. που δημιούργησε ένα μοντέλο για τη μελέτη των εκφράσεων του

προσώπου από μια ακολουθία εικόνων χρησιμοποιώντας 20 μεταβλητές παρακολούθησης. Η έρευνα διεξήχθη μέχρι το τέλος της δεκαετίας του 1980 και στις αρχές της δεκαετίας του 1990, επειδή έγινε διαθέσιμη, προηγμένη και πιο γρήγορη τεχνολογία υπολογιστών. Αυτό βοήθησε στην ανάπτυξη αλγορίθμων ανίχνευσης προσώπου και παρακολούθησης προσώπου στις αρχές της δεκαετίας του 1990. Ταυτόχρονα, ξεκίνησε και η έρευνα αλληλεπίδραση ανθρώπου-υπολογιστή (HCI).

3 Επισκόπηση WebRTC και Μηχανικής Μάθησης

3.1 Αρχές του WebRTC

Το WebRTC είναι ένα πρότυπο ανοιχτού κώδικα, που προσθέτει δυνατότητες επικοινωνίας σε πραγματικό χρόνο σε εφαρμογές ιστού και κινητά. Υποστηρίζει βίντεο, φωνή και γενικά δεδομένα που αποστέλλονται μεταξύ των συνομιλητών, επιτρέποντας ισχυρές λύσεις φωνητικής και οπτικοακουστικής επικοινωνίας. Η τεχνολογία είναι διαθέσιμη σε όλα τα σύγχρονα προγράμματα περιήγησης καθώς και σε εγγενείς εφαρμογές για όλες τις μεγάλες πλατφόρμες. Οι τεχνολογίες πίσω από το WebRTC διατίθενται ως API Javascript σε όλα τα μεγάλα προγράμματα περιήγησης και σε λειτουργικά Android και iOS. Η πρωτοβουλία WebRTC είναι ένα έργο που υποστηρίζεται κυρίως από την Apple, Google, Microsoft και Mozilla.

Το WebRTC είναι μια τεχνολογία που επιτρέπει σε εφαρμογές και ιστοτόπους να καταγράφουν και να προωθούν μέσα ήχου και βίντεο, καθώς και να ανταλλάσσουν αυθαίρετα δεδομένα μεταξύ των προγραμμάτων περιήγησης χωρίς να απαιτούν έναν διαμεσολαβητή. Το σύνολο των προτύπων που περιλαμβάνουν το WebRTC καθιστά δυνατή την κοινή χρήση δεδομένων και την εκτέλεση ομότιμης τηλεδιάσκεψης χωρίς να απαιτείται ο χρήστης να εγκαθιστά πρόσθετα ή οποιοδήποτε άλλο λογισμικό τρίτων κατασκευαστών.

Τα πρωτόκολλα επικοινωνίας, καθορίστηκαν από κοινού από τον W3C και την IETF, για να δημιουργήσουν ένα αξιόπιστο κανάλι επικοινωνίας μεταξύ των μελλοντικών προγραμμάτων περιήγησης ιστού. Η βασική ιδέα ήταν να καθοριστεί το API WebRTC το οποίο επιτρέπει την ασφαλή πρόσβαση στα περιφερειακά εισόδου, όπως τα μικρόφωνα και οι κάμερες web σε μια συσκευή, για να μοιράζονται ή να ανταλλάσσουν δεδομένα πολυμέσων, σε πραγματικό χρόνο με μια απομακρυσμένη συσκευή.

- Δεν χρειάζεται πρόσθετα: Το WebRTC είναι μια σύγχρονη τεχνολογία επικοινωνίας σε πραγματικό χρόνο χωρίς πρόσθετα. Δεν απαιτεί πρόσθετα ή εφαρμογές για ήχο, ροή βίντεο και κοινή χρήση δεδομένων. Χρησιμοποιεί Javascript διεπαφές

προγραμματισμού εφαρμογών και HTML5 για την ενσωμάτωση των τεχνολογιών επικοινωνίας στο πρόγραμμα περιήγησης. Προϊόντα όπως το Google Hangouts, το Whatsapp, το Facebook Messenger, η επικοινωνία της ομάδας ZOOM, η υποστήριξη πελατών της Zendesk, το Skype για το Web, έχουν ενσωματωμένο το WebRTC.

- Δουλεύει πάνω σε προγράμματα ιστού: Τα προγράμματα περιήγησης έχουν τη δυνατότητα άμεσης ανταλλαγής μέσων πραγματικού χρόνου με άλλα προγράμματα περιήγησης με τρόπο ομότιμης επικοινωνίας. Απλά η επίσκεψη στον ιστότοπο αρκεί για να εκκινήσει μια συνδιάλεξη μέσω του WebRTC
- Ασφάλεια: Προσφέρει υψηλό επίπεδο ασφάλειας από διάφορα άλλα συστήματα συνεχούς ροής, χωρίς την ανάγκη λογισμικού τρίτων.
- Είναι δωρεάν και χρησιμοποιείται ευρέως και παγκοσμίως.
- Peer to Peer: Το WebRTC υποστηρίζει την σύνδεση ομότιμων χρηστών.

3.2 Μηχανική Μάθηση

Μηχανική μάθηση είναι μια επιστήμη, ένα υποσύνολο τεχνητής νοημοσύνης, που με την αναγνώριση προτύπων και την υπολογιστική θεωρία μάθησης, μπορεί να κατασκευάσει αλγορίθμους από δεδομένα

Τεχνητά Νευρωνικά δίκτυα είναι αλγόριθμοι εκμάθησης εμπνευσμένοι από την δομή και την λειτουργία των βιολογικών νευρωνικών δικτύων. Χρησιμοποιούν μια σειρά από εσωτερικά φίλτρα για την επεξεργασία δεδομένων, έτσι ώστε να εξάγουν πληροφορία για την συσχέτιση μεταξύ δεδομένων εισόδου εξόδου, την ανακάλυψη προτύπων ή την στατιστική δομή όταν αυτά βρίσκονται πάνω σε μια άγνωστη κατανομή πιθανότητας. Ο τρόπος δόμησης των εσωτερικών νευρώνων καθορίζουν και το είδος τους νευρωνικού δικτύου:

- DNN (Deep Neural Netowrks): Ανάλογα με τον αριθμό των νευρώνων τα νευρωνικά δίκτυα χωρίζονται σε ρηχά και βαθιά. Επειδή η τεχνολογία των μικροεπεξεργαστών έχει μικρό κόστος, έχει προχωρήσει το μοντέλο των βαθιών νευρωνικών δικτύων, καθόσον φέρουν μεγαλύτερη απόδοση. Ένα βαθύ νευρωνικό δίκτυο (DNN) είναι ένα τεχνητό νευρωνικό δίκτυο (ANN) με πολλαπλά επίπεδα μεταξύ των δεδομένων εισόδου και εξόδου.

- Συνελικτικό Νευρωνικό Δίκτυο : Στη βαθιά εκμάθηση, ένα συνελικτικό νευρωνικό δίκτυο (Συνελικτικό Νευρωνικό Δίκτυο, ή ConvNet) είναι μια κλάση βαθιά νευρωνικών δικτύων, που συνήθως εφαρμόζεται στην ανάλυση οπτικών εικόνων. Είναι επίσης γνωστά ως αμετάβλητα τεχνητά νευρωνικά δίκτυα μεταβλητής (SIANN). Έχουν εφαρμογές στην αναγνώριση εικόνων και βίντεο, συστηματοποιημένα συστήματα, ταξινόμηση εικόνων, ανάλυση ιατρικών εικόνων, επεξεργασία φυσικής γλώσσας και οικονομικές χρονολογικές σειρές.
- RNN (Recursive Neural Network): Ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) είναι μια τάξη τεχνητών νευρωνικών δικτύων όπου οι συνδέσεις μεταξύ των κόμβων σχηματίζουν ένα κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής αλληλουχίας. Αυτό του επιτρέπει να παρουσιάζει χρονική δυναμική συμπεριφορά. Προέρχονται από τα ανεστραμμένα νευρωνικά δίκτυα. Τα RNNs μπορούν να χρησιμοποιήσουν την εσωτερική τους κατάσταση (μνήμη) για να επεξεργαστούν ακολουθίες μεταβλητού μήκους εισόδων. Αυτό το καθιστά εφαρμόσιμο σε εργασίες όπως η μη κατακερματισμένη, συνδεδεμένη αναγνώριση χειρόγραφου ή την αναγνώριση ομιλίας.

3.3 Αναγνώριση συναισθημάτων σε πραγματικό χρόνο

Για την αναγνώριση συναισθημάτων, χρησιμοποιήθηκαν μεθοδολογίες μηχανικής μάθησης για την απομόνωση και εξαγωγή του προσώπου από την εικόνα, την αναγνώριση χαρακτηριστικών και την ταξινόμησή τους μέσα από προ-εκπαιδευμένα μοντέλα μηχανικής μάθησης. Οι αλγόριθμοι και τα μοντέλα, σχεδιάστηκαν με τρόπο που να δίνουν την μέγιστη δυνατή ευστοχία με την ελάχιστη κατανάλωση πόρων. Οι τεχνολογίες και οι αλγόριθμοι είναι μοντελοποιημένοι, για να λειτουργήσουν σε εμπορικές συσκευές (φορητοί υπολογιστές και κινητά) που χρησιμοποιούνται συνηθέστερα.

3.4 Λειτουργικότητα WebRTC σε προγράμματα περιηγητή

Το WebRTC επιτρέπει την επικοινωνία μεταξύ των προγραμμάτων περιήγησης. Μια εφαρμογή ιστού WebRTC προγραμματίζεται ως ένα μείγμα HTML και Javascript.

Λειτουργεί και επικοινωνεί με τα προγράμματα περιήγησης ιστού μέσω του τυποποιημένου API. Επομένως, το WebRTC API πρέπει να παρέχει μια σειρά βοηθητικών βιβλιοθηκών. Ορισμένες από αυτές αφορούν τη διαχείριση των συνδέσεων (σε επίπεδο peer-to-peer), τη διαπραγμάτευση, την επιλογή και τον έλεγχο των δυνατοτήτων κωδικοποίησης / αποκωδικοποίησης των δεδομένων ήχου και βίντεο, τον έλεγχο μέσων και το τείχος προστασίας.

3.5 Ο πυρήνας του WebRTC χωρίζεται σε τρία στοιχεία

- **MediaStream:** Το πρώτο βήμα αποτελείται από τον τύπο των δεδομένων που ο χρήστης θέλει να μοιραστεί. Σε αυτήν την περίπτωση, η ροή που θέλει ο χρήστης (ήχου / βίντεο) επιλέγεται και χρησιμοποιείται ο ανάλογος τρόπος επικοινωνίας που πρέπει να δημιουργηθεί. Η ροή τοπικών μέσων εκχωρεί στο πρόγραμμα περιήγησης πρόσβαση σε συσκευές ροής, όπως η κάμερα και το μικρόφωνο. Επιτρέπει επίσης στο πρόγραμμα περιήγησης να καταγράψει από τις συσκευές πολυμέσων. Ένας χρήστης μπορεί να χρησιμοποιήσει τη λειτουργία `getUserMedia` για να πάρει την πρόσβαση από το πρόγραμμα περιήγησης.
- **RTCPeerConnection:** Μόλις ο χρήστης επιλέξει τη ροή επικοινωνίας τότε το επόμενο βήμα είναι να συνδεθεί με το σύστημα του συνομιλητή. Επιτρέπει στο πρόγραμμα περιήγησης να ανταλλάσσει δεδομένα απευθείας με προγράμματα περιήγησης συνομιλητών για κλήσεις φωνής και βίντεο. Επιπλέον, επιτρέπει τη σύνδεση μεταξύ του αποστολέα και του δέκτη μέσω διακομιστών STUN και TURN.
- **RTCDataChannel:** Δίνει την δυνατότητα στα προγράμματα περιήγησης να ανταλλάσσουν δεδομένα με αμφίδρομο peer-to-peer. Η λειτουργία `CreateDataChannel` καλείται για πρώτη φορά σε ένα επιλεγμένο αντικείμενο `PeerConnection`.

3.6 Το WebRTC χωρίζεται σε τρία διαφορετικά επίπεδα API

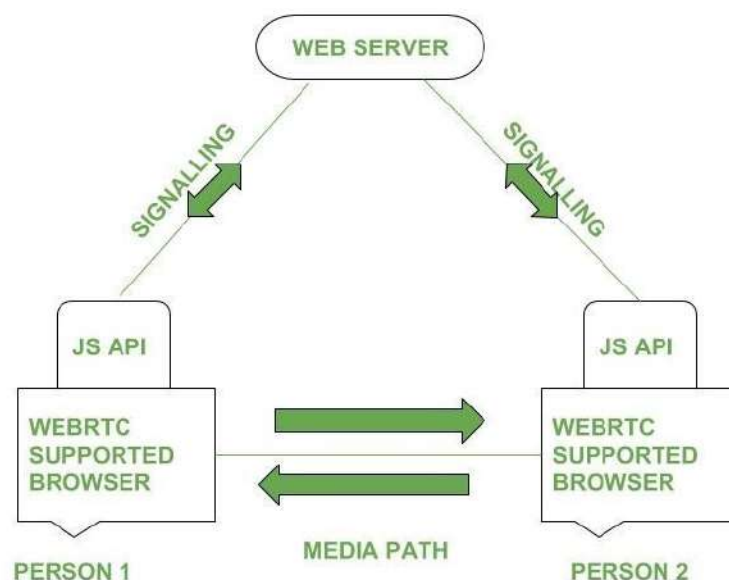
- Το πρώτο επίπεδο περιέχει όλες τις απαιτήσεις των APIs για τον προγραμματισμό των εφαρμογών διαδικτύου, συμπεριλαμβανομένων των αντικειμένων `RTCPeerConnection`, `RTCDataChannel` και `streaming media` και των λειτουργιών

τους.

- Δεύτερο είναι το API για τους κατασκευαστές προγραμμάτων περιήγησης.
- Το τρίτο είναι το Overridable API, το οποίο οι κατασκευαστές περιηγητών δημιουργούν και αναπτύσσουν.

3.7 Βασική επικοινωνία μέσω WebRTC

Αρχικά σε ένα σενάριο, δύο χρήστες συνδέονται σε μια εφαρμογή WebRTC. Στην συνέχεια θέλει ο ένας να επικοινωνήσει με τον άλλον. Εκεί χρησιμοποιούνται τα μηνύματα σηματοδότησης των οποίων η εργασία είναι να ρυθμίσουν και να τερματίσουν τις επικοινωνίες. Αυτά τα μηνύματα μεταφέρονται σε διακομιστές ιστού από το πρωτόκολλο HTTP ή WebSocket το οποίο μπορεί να τις τροποποιήσει, να μεταφράσει ή να τις διαχειριστεί όπως απαιτείται. Όσον αφορά τη διαδρομή δεδομένων, ένα PeerConnection επιτρέπει τη ροή μέσων απευθείας μεταξύ των προγραμμάτων περιήγησης χωρίς παρεμβαλλόμενους διακομιστές. Οι δύο διακομιστές ιστού μπορούν να επικοινωνούν χρησιμοποιώντας ένα τυποποιημένο πρωτόκολλο σηματοδότησης και έτσι δημιουργείται η επικοινωνία.



Σχήμα 3.1: Τυπική επικοινωνία μέσω WebRTC μέσω signaling server

3.8 Επικοινωνία Peer-To-Peer

Το WebRTC μπορεί να χρησιμοποιηθεί για πολλαπλές εφαρμογές, αλλά το κύριο πλεονέκτημα είναι η επικοινωνία ήχου και βίντεο σε πραγματικό χρόνο μέσω ενός προγράμματος περιήγησης ιστού. Ο περιηγητής ιστού κάθε χρήστη πρέπει να συμφωνήσει να ξεκινήσει την επικοινωνία, να ξέρει πώς να εντοπίσει ο ένας τον άλλον, να παρακάμπτει την ασφάλεια και την προστασία του τείχους προστασίας και να μεταδίδει όλες τις επικοινωνίες πολυμέσων σε πραγματικό χρόνο. Μία από τις μεγαλύτερες προκλήσεις που σχετίζονται με τις επικοινωνίες peer-to-peer με βάση το πρόγραμμα περιήγησης είναι η γνώση του πώς να εντοπιστεί και να δημιουργηθεί μια σύνδεση υποδοχής δικτύου με το πρόγραμμα περιήγησης ιστού άλλου υπολογιστή, προκειμένου να μεταδώσει αμφίδρομα τα δεδομένα πολυμέσων. Οι υπολογιστές των περισσότερων χρηστών βρίσκονται συνήθως πίσω από μια συσκευή μετάφρασης τείχους προστασίας και πρόσβασης δικτύου (NAT) και ως εκ τούτου δεν του έχει εκχωρηθεί μια στατική δημόσια διεύθυνση IP. Μια συσκευή

NAT μεταφράζει ιδιωτικές διευθύνσεις IP από το εσωτερικό ενός τείχους προστασίας σε δημόσιες διευθύνσεις IP. Απαιτούνται συσκευές NAT για περιορισμούς ασφαλείας και IPv4 σε διαθέσιμες δημόσιες διευθύνσεις IP. Αυτό καλείται να λύσει το WebRTC Firewall και το NAT Traversal.

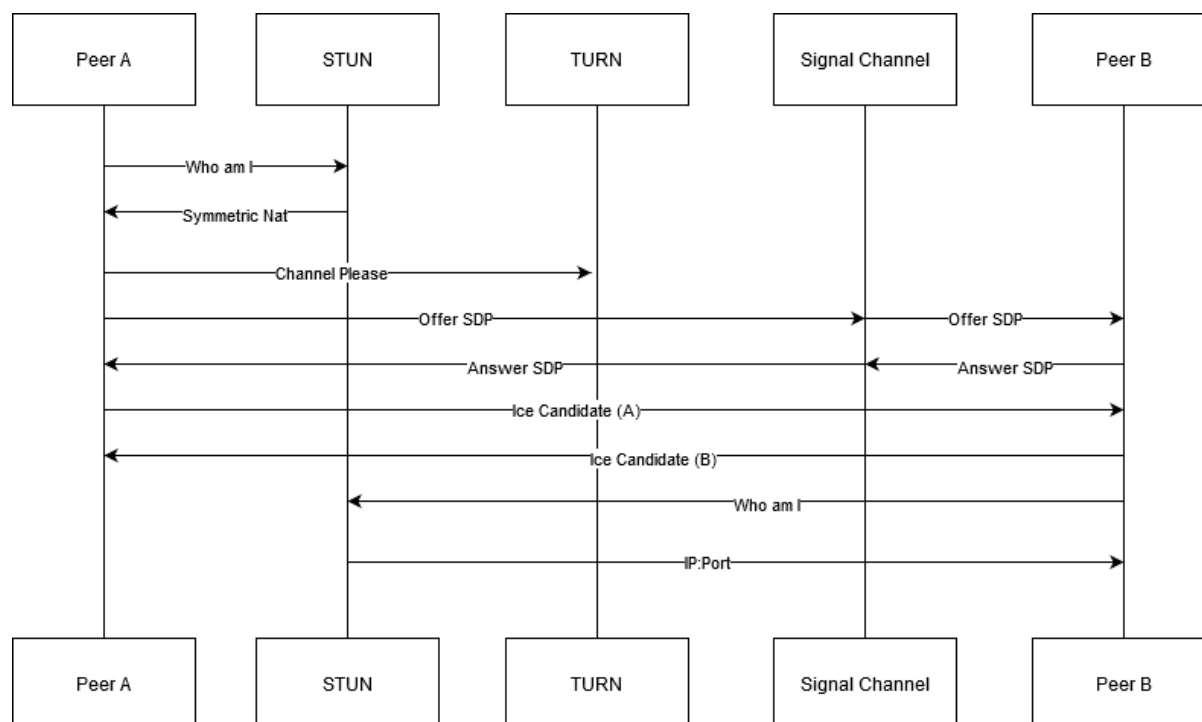
3.9 Σηματοδοσία, περίοδος σύνδεσης και πρωτόκολλα

Η διαδικασία ανίχνευσης πληροφοριών δικτύου που περιγράφεται παραπάνω είναι ένα μέρος του μεγαλύτερου θέματος σηματοδότησης που στην περίπτωση του WebRTC, βασίζεται στο πρότυπο πρωτόκολλο εγκατάστασης του Javascript Session Establishment Protocol (JSEP). Η σηματοδότηση περιλαμβάνει την ανακάλυψη δικτύου και τη μετάδοση NAT, τη δημιουργία και τη διαχείριση των συνεδριών, την ασφάλεια επικοινωνίας, τα μεταδεδομένα και τον συντονισμό των δυνατοτήτων των μέσων και τον χειρισμό σφαλμάτων.

Η σηματοδότηση δεν καθορίζεται από το πρότυπο WebRTC ούτε εφαρμόζεται από τα API του ώστε να επιτρέπεται ευελιξία στις τεχνολογίες και τα πρωτόκολλα που χρησιμοποιούνται. Η σηματοδότηση και ο διακομιστής που τη διαχειρίζεται αφήνεται στον δημιουργό της εφαρμογής WebRTC να το λύσει, επειδή είναι σε θέση να προσδιορίσει τη διεύθυνση IP που χρησιμοποιεί ο διακομιστής STUN. Το επόμενο βήμα είναι να διαπραγματευτεί και να δημιουργήσει τη σύνδεση σύνδεσης δικτύου με τον συνομιλητή. Αυτή η διαδικασία είναι ανάλογη με την πραγματοποίηση τηλεφωνικής κλήσης. Η αρχική διαπραγμάτευση και σύσταση της σύσκεψης γίνεται χρησιμοποιώντας ένα πρωτόκολλο σηματοδότησης / επικοινωνίας που ειδικεύεται στις επικοινωνίες πολυμέσων. Το πρωτόκολλο αυτό είναι επίσης υπεύθυνο για τη ρύθμιση των κανόνων με τους οποίους η συνεδρία διοικείται και τερματίζεται.

Το πρωτόκολλο έναρξης σύνδεσης (SIP) και το πρωτόκολλο περιγραφής συνόλων (SDP) είναι τα συνηθέστερα που χρησιμοποιούνται στην περίπτωση του WebRTC. Όλα τα μεταδεδομένα, ειδικά για πολυμέσα περνούν χρησιμοποιώντας το πρωτόκολλο SDP. Οποιοσδήποτε ομότιμος χρήστης που επιχειρεί να επικοινωνήσει με έναν άλλο ομότιμο χρήστη, δημιουργεί ένα σύνολο υποψηφίων ICE, όπου το ICE αντιπροσωπεύει το πρωτόκολλο Interactive Connectivity Establishment. Οι υποψήφιοι αντιστοιχίζονται σε έναν δεδομένο συνδυασμό διευθύνσεων IP και πρωτόκολλο μεταφοράς που πρόκειται να

χρησιμοποιήσουν.



Σχήμα 3.2: Σήματα ανταλλαγής πληροφορίας για την επικοινωνία δύο χρηστών

Κάθε ομότιμος χρήστης καθιερώνει πρώτα τη διεύθυνση IP του κοινού, όπως περιγράφεται. Τα "κανάλια" για δεδομένα σήματος δημιουργούνται στη συνέχεια δυναμικά για να ανιχνεύουν συνομιλητές και να υποστηρίξουν τις διαπραγματεύσεις μεταξύ ομότιμων και τη σύνοδο. Ένας ομιλητής στέλνει μια "προσφορά" χρησιμοποιώντας ένα πρωτόκολλο σηματοδότησης SDP. Περιμένει να λάβει μια "απάντηση", από οποιονδήποτε δέκτη που είναι συνδεδεμένος με το δεδομένο "κάνάλι". Μόλις ληφθεί η απάντηση, γίνεται μια διαδικασία για τον προσδιορισμό και τη διαπραγμάτευση των καλύτερων υποψηφίων του ICE που συγκεντρώνονται από κάθε ομότιμο. Μόλις επιλεγούν οι βέλτιστοι υποψήφιοι ICE, συμφωνούνται ουσιαστικά όλα τα απαιτούμενα μεταδεδομένα, η δρομολόγηση δικτύου (διεύθυνση IP και θύρα) και οι πληροφορίες μέσων που χρησιμοποιούνται για την επικοινωνία για κάθε συνομιλητή. Η σύνοδος υποδοχής δικτύου μεταξύ των συνομιλητών είναι τότε πλήρως καθιερωμένη και ενεργή. Στη συνέχεια, οι τοπικές ροές δεδομένων και τα τελικά σημεία του καναλιού δεδομένων δημιουργούνται από κάθε ομότιμο χρήστη και τα δεδομένα πολυμέσων μεταδίδονται τελικά με τους δύο τρόπους χρησιμοποιώντας οποιαδήποτε αμφίδρομη τεχνολογία επικοινωνίας.

Εάν αποτύχει η διαδικασία συμφωνίας για τον καλύτερο υποψήφιο ICE, κάτι

που συμβαίνει μερικές φορές λόγω τείχους προστασίας και τεχνολογιών NAT που χρησιμοποιούνται, η εφευρισκή λύση είναι να χρησιμοποιηθεί ένας διακομιστής TURN ως ρελέ. Αυτή η διαδικασία χρησιμοποιεί ένα διακομιστή που ενεργεί ως ενδιάμεσος και ο οποίος αναμεταδίδει οποιαδήποτε μεταδιδόμενα δεδομένα μεταξύ συνομιλητών. Αυτό έρχεται σε αντίθεση με την πραγματική επικοινωνία ομότιμης αλληλεπίδρασης, στην οποία και οι δύο ομότιμοι διαβιβάζουν αμφίδρομα τα δεδομένα απευθείας μεταξύ τους. Όταν χρησιμοποιείται την εναλλακτική λύση TURN για επικοινωνίες, κάθε ομότιμος χρήστης δεν χρειάζεται πλέον να γνωρίζει πώς να επικοινωνεί και να μεταδίδει δεδομένα μεταξύ τους. Αντίθετα, πρέπει να γνωρίζουν ποιο δημόσιο διακομιστή TURN αποστέλλει και λαμβάνει δεδομένα πολυμέσων σε πραγματικό χρόνο κατά τη διάρκεια μιας συνεδρίας επικοινωνίας. Οι διακομιστές TURN πρέπει να είναι αρκετά ανθεκτικοί, να έχουν εκτεταμένο εύρος ζώνης και δυνατότητες επεξεργασίας και να χειρίζονται δυνητικά μεγάλα ποσά δεδομένων. Επομένως, η χρήση ενός διακομιστή TURN προφανώς συνεπάγεται πρόσθετο κόστος και πολυπλοκότητα.

3.9.1 NAT

Η μετάφραση διεύθυνσης δικτύου (NAT) είναι ένας μηχανισμός που αποκρύπτει από την πρόσβαση του κοινού τις ιδιωτικές διευθύνσεις IP των μηχανών μέσα σε ένα δίκτυο. Αυτός ο μηχανισμός NAT βρίσκονται συνήθως σε όλους τους τύπους συσκευών δικτύου, που κυμαίνονται από οικιακούς δρομολογητές μέχρι ολοκληρωμένες εταιρικές τράπεζες τείχους προστασίας. Σε όλες τις περιπτώσεις το αποτέλεσμα είναι το ίδιο: οι μηχανές μέσα στο NAT δεν μπορούν να έχουν ελεύθερη πρόσβαση από το εξωτερικό.

Τα αποτελέσματα ενός NAT είναι πολύ αρνητικά για τις επικοινωνίες WebRTC: οι μηχανές μέσα στο δίκτυο θα μπορούν να στέλνουν δεδομένα προς το εξωτερικό, αλλά δεν θα μπορούν να λαμβάνουν δεδομένα από απομακρυσμένα τελικά σημεία που υπάρχουν έξω από το δίκτυο. Προκειμένου να επιτρέπεται αυτή η δυνατότητα, οι συσκευές NAT συνήθως επιτρέπουν τη διαμόρφωση των συνδέσεων NAT για να επιτρέπουν την είσοδο δεδομένων από το εξωτερικό μέρος του δικτύου. Η δημιουργία αυτών των δεσμών NAT είναι αυτό που ονομάζεται NAT traversal, επίσης κοινώς αναφερόμενο ως "άνοιγμα θυρών".

3.9.2 ICE

Η δημιουργία αλληλεπιδραστικής σύνδεσης (ICE) είναι ένα πρωτόκολλο που χρησιμοποιείται για την μετάβαση NAT. Ορίζει μια τεχνική που επιτρέπει την επικοινωνία μεταξύ δύο τελικών σημείων όταν κάποιος είναι μέσα σε NAT και ο άλλος είναι έξω από αυτό. Το καθαρό αποτέλεσμα της διαδικασίας ICE είναι ότι το NAT θα μείνει με όλες τις απαραίτητες θύρες ανοιχτές για επικοινωνία και τα δύο τελικά σημεία θα έχουν πλήρη πληροφορία σχετικά με τη διεύθυνση IP και τις θύρες όπου μπορεί να έρθει σε επαφή το άλλο τελικό σημείο. Το ICE δεν λειτουργεί αυτόνομα: πρέπει να χρησιμοποιήσει ένα πρωτόκολλο βοήθειας που ονομάζεται STUN.

3.9.3 STUN

Το Traversal Utilities for Session (STUN) είναι ένα πρωτόκολλο που συμπληρώνει το ICE στο έργο της επίλυσης του προβλήματος NAT traversal. Μπορεί να χρησιμοποιηθεί από οποιεσδήποτε παραμέτρους για τον προσδιορισμό της διεύθυνσης IP και της θύρας που έχει εκχωρηθεί από ένα NAT. Μπορεί επίσης να χρησιμοποιηθεί για να ελέγξει τη συνδεσιμότητα μεταξύ δύο τελικών σημείων και ως ένα πρωτόκολλο για να διατηρήσει ζωντανές τις συνδέσεις NAT. Το STUN λειτουργεί με πολλά υπάρχοντα NAT και δεν απαιτεί ιδιαίτερη συμπεριφορά από αυτά.

3.9.4 TURN

Η χρήση των ρελέδων γύρω από το NAT (TURN) είναι μια επέκταση του STUN, που χρησιμοποιείται όπου οι πολιτικές ασφαλείας είναι NAT. Είναι πολύ κρίσιμες και οι απαραίτητες συνδέσεις, όταν οι συνδέσεις μέσω NAT δεν μπορούν να δημιουργηθούν με επιτυχία. Σε αυτές τις περιπτώσεις, είναι απαραίτητο για τον κεντρικό υπολογιστή να χρησιμοποιεί τις υπηρεσίες ενός ενδιάμεσου κόμβου που λειτουργεί ως ρελές επικοινωνίας.

3.9.4.1 Πότε χρειάζεται STUN

STUN απαιτείται για κάθε τελικό σημείο πίσω από NAT. Όλοι οι συνομιλητές NAT πρέπει να ανοίξουν τις δικές τους θύρες, κάνοντας NAT traversal, χρησιμοποιώντας ένα διακομιστή STUN που βρίσκεται εκτός του NAT. Ομοίως, όλα τα τελικά σημεία του προγράμματος περιήγησης που βρίσκονται πίσω από ένα NAT πρέπει να διαμορφώσουν

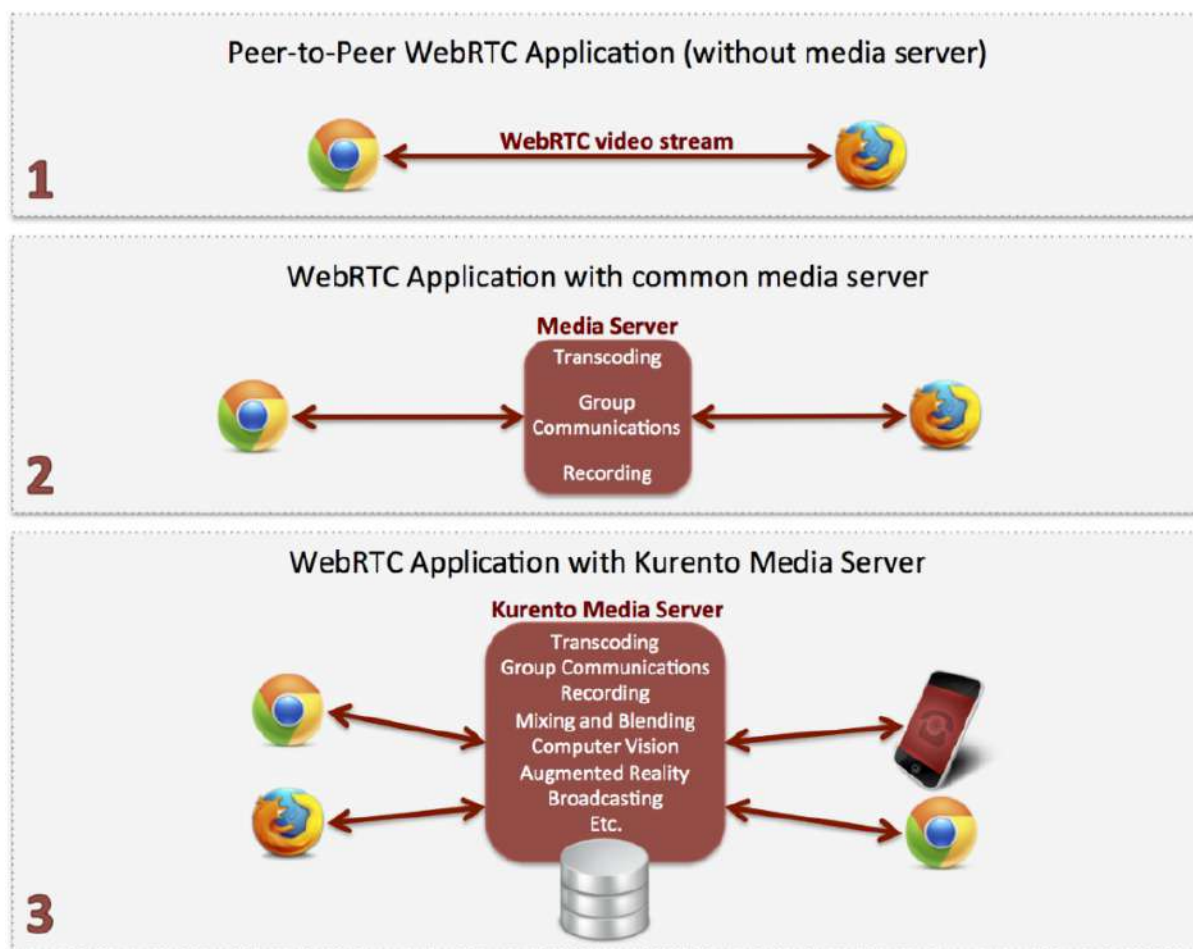
τις λεπτομέρειες διακομιστή STUN και / ή TURN με το πεδίο `iceServers` του κατασκευαστή `RTCPeerConnection`.

3.10 WebRTC Javascript API

Το WebRTC είναι ένα περίπλοκο πρωτόκολλο με πολλές τεχνολογίες που εμπλέκονται. Οι διαδικασίες που περιγράφονται υλοποιούνται μέσω ενός συνόλου API Javascript που παράγουν και μεταδίδουν τα δεδομένα πολυμέσων που χρησιμοποιούνται για επικοινωνίες σε πραγματικό χρόνο. Το WebRTC επιτρέπει σε μια εφαρμογή βασισμένη στον περιηγητή ιστού του υπολογιστή ή στον περιηγητή ιστού του κινητού να έχει πρόσβαση στο μικρόφωνο και την κάμερα. Το πρόγραμμα περιήγησης ενημερώνει τον χρήστη ότι μια εφαρμογή ζητά πρόσβαση στην κάμερα και το μικρόφωνο του υπολογιστή. Μόλις ο χρήστης επιτρέψει την πρόσβαση για να χρησιμοποιήσει αυτές τις συσκευές, το WebRTC μπορεί να δημιουργήσει μεμονωμένα ρεύματα μεταδίδσιμων δεδομένων ήχου και βίντεο από τα δεδομένα που παράγονται από αυτές τις συσκευές εισόδου. Αυτά τα δεδομένα διαβιβάζονται έπειτα μέσω καναλιών δεδομένων δικτύου που δημιουργήθηκαν από τις διαδικασίες που αναφέρθηκαν προηγουμένως. Τα κύρια WebRTC API περιλαμβάνουν το `Navigator.getUserMedia` (λήψη ήχου και βίντεο), το `RTCPeerConnection` (δημιουργία και διαπραγμάτευση συνδέσεων peer-to-peer) και το `RTCDataChannel` (αντιπροσωπεύει ένα αμφίδρομο κανάλι δεδομένων μεταξύ συνομιλητών).

3.11 Kurento Framework

Το Kurento είναι ένας διακομιστής πολυμέσων WebRTC και ένα σύνολο API-clients που απλοποιούν την ανάπτυξη προηγμένων εφαρμογών βίντεο για πλατφόρμες world wide web και smartphone. Οι λειτουργίες του Kurento Media Server περιλαμβάνουν ομαδικές επικοινωνίες, μεταγλώττιση, εγγραφή, ανάμειξη, εκπομπή και δρομολόγηση οπτικοακουστικών ροών. Το Kurento Media Server παρέχει επίσης προηγμένες δυνατότητες επεξεργασίας πολυμέσων που περιλαμβάνουν όραση υπολογιστή, ευρετηρίαση βίντεο, επαυξημένη πραγματικότητα και ανάλυση ομιλίας. Η αρθρωτή αρχιτεκτονική του Kurento, καθιστά απλή την ενσωμάτωση αλγορίθμων επεξεργασίας πολυμέσων τρίτων όπως αναγνώριση ομιλίας, ανάλυση συναισθημάτων και αναγνώριση προσώπου, μια βατή διαδικασία.



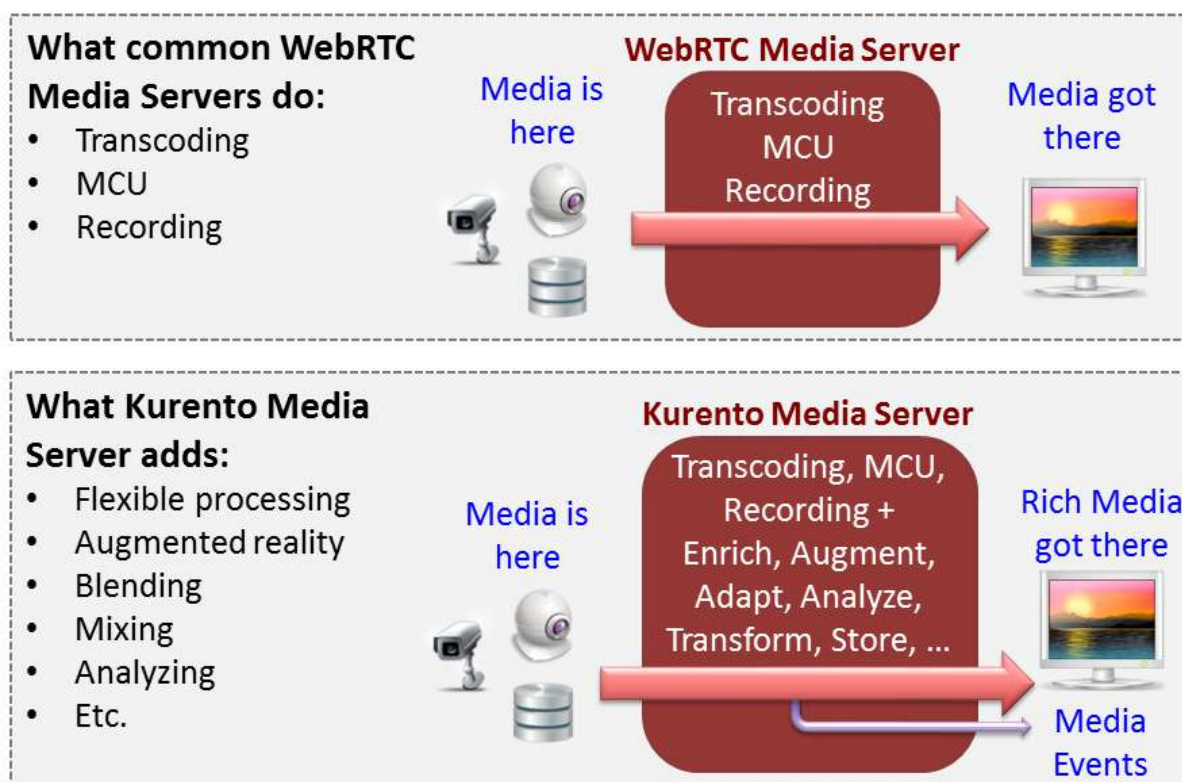
Σχήμα 3.3: Σενάρια επικοινωνίας μεταξύ διαφορετικών πελατών

3.12 Kurento Media Server

Το κύριο συστατικό του Kurento, είναι το Kurento Media Server (KMS), που είναι υπεύθυνο για τη μετάδοση, επεξεργασία, εγγραφή και αναπαραγωγή των πολυμέσων. Το KMS είναι χτισμένο στην κορυφή της βιβλιοθήκης πολυμέσων GStreamer και παρέχει τις ακόλουθες δυνατότητες:

1. Διαδικτυακά πρωτόκολλα ροής δεδομένων, συμπεριλαμβανομένων των HTTP, RTP και WebRTC.
2. Ομαδικές επικοινωνίες (λειτουργικότητα MCU και SFU) που υποστηρίζουν τόσο την ανάμειξη πολυμέσων όσο και τη δρομολόγηση / αποστολή πολυμέσων.
3. Γενική υποστήριξη για φίλτρα που υλοποιούν αλγόριθμους Computer Vision και Augmented Reality.

4. Αποθήκευση πολυμέσων που υποστηρίζει γραπτές λειτουργίες για WebM και MP4 και παίζει σε όλες τις μορφές που υποστηρίζει το GStreamer.
5. Αυτόματη κρυπτογράφηση πολυμέσων μεταξύ οποιουδήποτε από τους κωδικοποιητές που υποστηρίζονται από το GStreamer, συμπεριλαμβανομένων των VP8, H.264, H.263, AMR, OPUS, Speex και G.711.



Σχήμα 3.4: Ενθυλάκωση λειτουργιών του WebRTC και ενσωμάτωση καινούργιων

3.13 Αρχές σχεδιασμού του Kurento

3.13.1 Ξεχωριστά στάδια πολυμέσων και σηματοδότησης

Η σηματοδότηση και τα πολυμέσα είναι δύο ξεχωριστά επίπεδα που το Kurento τα χειρίζεται ξεχωριστά.

3.13.2 Κατανομή πολυμέσων και υπηρεσιών εφαρμογών

Ο διακομιστής Kurento Media Server και οι εφαρμογές του μπορούν να επικεντρωθούν, κλιμακωθούν ή διανεμηθούν μεταξύ διαφορετικών μηχανών. Μια ενιαία εφαρμογή μπορεί

να επικαλεσθεί τις υπηρεσίες περισσότερων από έναν διακομιστή Media Server του Kurento. Το αντίθετο επίσης ισχύει, δηλαδή ένας διακομιστής Kurento Media Server μπορεί να παρακολουθήσει τις αιτήσεις για περισσότερες από μία εφαρμογές.

3.13.3 Κατάλληλο για το σύννεφο

Το Kurento είναι κατάλληλο για να ενσωματωθεί σε περιβάλλοντα σύννεφων για να λειτουργήσει ως στοιχείο PaaS (πλατφόρμα ως υπηρεσία).

3.13.4 Αγωγοί Πολυμέσων

Η αλυσίδα στοιχείων πολυμέσων μέσω των αγωγών πολυμέσων είναι μια προσέγγιση για την απλοποίηση της πολυπλοκότητας της επεξεργασίας πολυμέσων.

3.13.5 Ανάπτυξη εφαρμογής

Οι προγραμματιστές δεν χρειάζεται να γνωρίζουν την εσωτερική πολυπλοκότητα του Kurento Media Server: όλες οι εφαρμογές μπορούν να αναπτυχθούν σε οποιαδήποτε τεχνολογία ή πλαίσιο που επιθυμεί ο προγραμματιστής, από πελάτη σε διακομιστή. Από τα προγράμματα περιήγησης έως υπηρεσίες cloud.

3.13.6 Δυνατότητα επικοινωνίας από άκρο σε άκρο

Το Kurento παρέχει δυνατότητες επικοινωνίας από άκρο σε άκρο, έτσι ώστε οι προγραμματιστές δεν χρειάζεται να αντιμετωπίζουν την πολυπλοκότητα της μεταφοράς, κωδικοποίησης / αποκωδικοποίησης και αποτύπωσης πολυμέσων σε συσκευές πελάτη.

3.13.7 Πλήρως επεξεργάσιμα ρεύματα πολυμέσων

Το Kurento επιτρέπει όχι μόνο αλληλεπιδραστικές σε διαπροσωπικές επικοινωνίες, αλλά και από άνθρωπο σε μηχανή (π.χ. Video on Demand μέσω ροής δεδομένων σε πραγματικό χρόνο) και από μηχανή σε μηχανή (π.χ. ανταλλαγή δεδομένων πολλαπλών σημάτων).

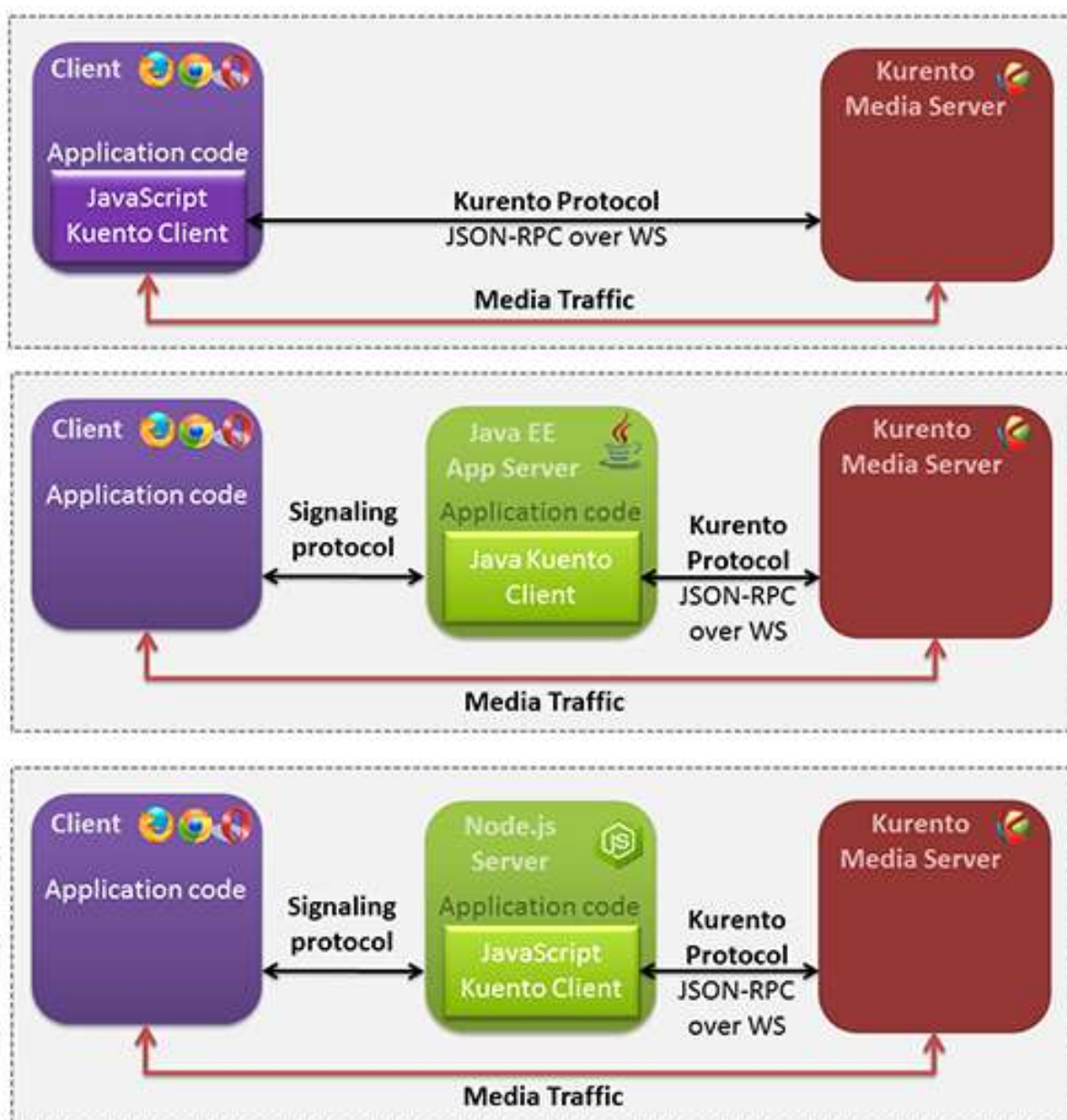
3.13.8 Υψηλό επίπεδο προσαρμογής πολυμέσων

Το Kurento παρέχει ένα υψηλό επίπεδο προσαρμογής πολυμέσων, για να καταστεί δυνατή η επικοινωνία μεταξύ διαφορετικών συσκευών που έχουν διαφορετικές απαιτήσεις όπως το μέγεθος της οθόνης, την κατανάλωση ενέργειας, και τον ρυθμό μετάδοσης.

3.14 Χαρακτηριστικά του Kurento

3.14.1 Kurento API

Είναι ο διακομιστής που χειρίζεται λειτουργίες χαμηλού επιπέδου για την μετάδοση ροής πολυμέσων. Ο διακομιστής Kurento Media Server εκθέτει όλες τις λειτουργίες του μέσω API RPC που ονομάζεται API Kurento. Το kurento client API είναι έτοιμο προς χρήση με το Kurento Media Server. Αυτά παρέχονται αυτήν τη στιγμή για Java, Browser Javascript και Node.js.

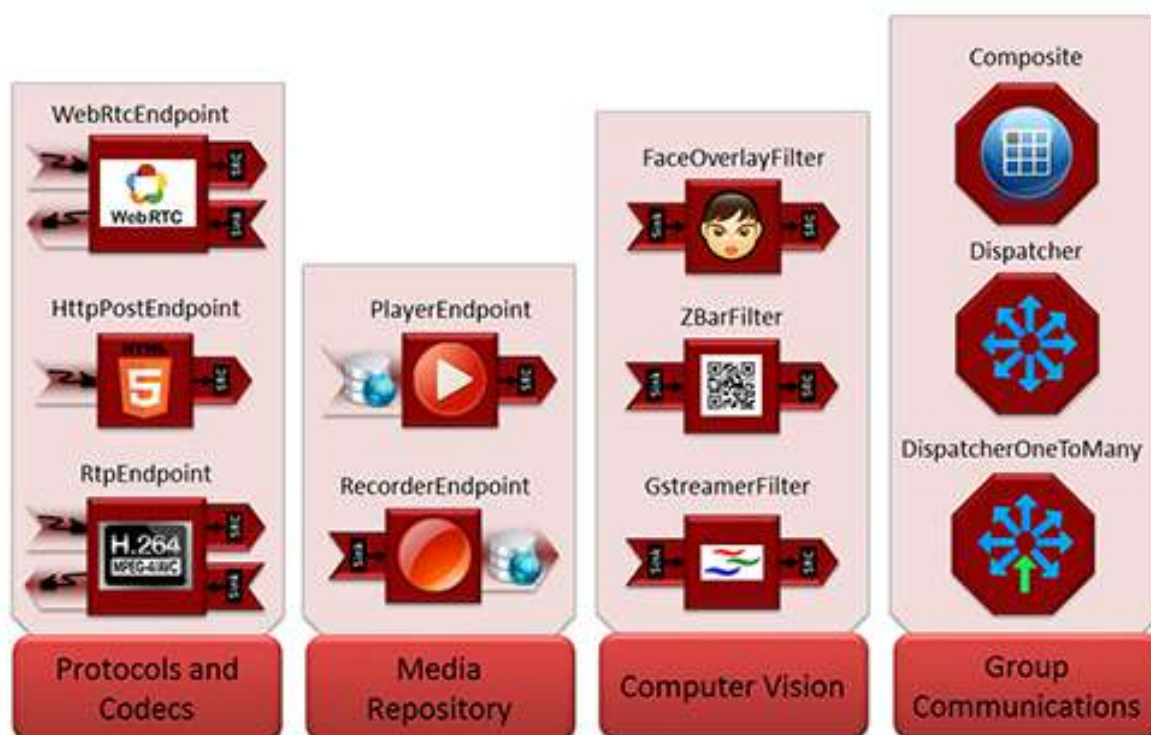


Σχήμα 3.5: Παράδειγμα του Kurento Clients σε τρία σενάρια: 1.Χρήση του Client Kurento Java σε διακομιστή εφαρμογών Java EE. 2.Χρησιμοποιώντας το πρόγραμμα-πελάτη Javascript Kurento σε ένα διακομιστή Node.js. 3.Σύνδεση πελατών Kurento (Java και Javascript) στο Kurento Media Server

3.14.2 Kurento API client

Το Kurento API Client βασίζεται στην έννοια των στοιχείων πολυμέσων. Ένα στοιχείο πολυμέσων διαθέτει μια συγκεκριμένη δυνατότητα πολυμέσων. Για παράδειγμα, το στοιχείο πολυμέσων που ονομάζεται **WebRTCEndpoint** διατηρεί τη δυνατότητα αποστολής και λήψης ροών πολυμέσων WebRTC. το στοιχείο πολυμέσων που ονομάζεται **RecorderEndpoint** έχει τη δυνατότητα εγγραφής στο σύστημα αρχείων οποιωνδήποτε ροών

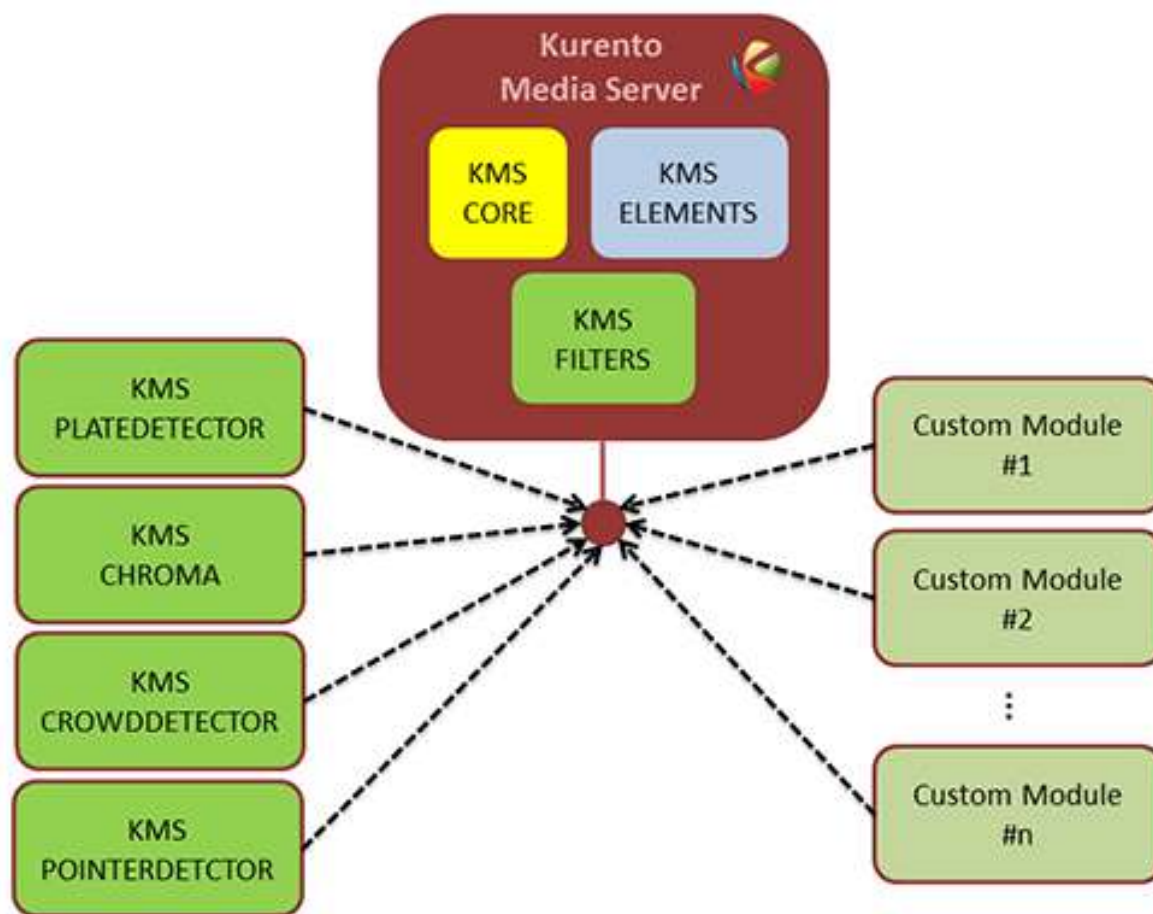
πολυμέσων που λαμβάνει. Το FaceOverlayFilter ανιχνεύει πρόσωπα στις ανταλλασσόμενες ροές βίντεο και προσθέτει μια συγκεκριμένη επικαλυπτόμενη εικόνα πάνω από αυτά.



Σχήμα 3.6: Στοιχεία πολυμέσων του Kurento

3.14.3 Τμήματα του Kurento

Το Kurento είναι προσαρμόσιμη λύση. Ο διακομιστής Kurento Media Server χρησιμοποιεί από προεπιλογή διάφορα τμήματα, με όνομα kms-core, kms-elements και kms-filters. Επιπλέον, υπάρχουν και άλλες ενσωματωμένες λειτουργικότητες για την ενίσχυση των δυνατοτήτων που παρέχονται από τον διακομιστή Kurento Media Server. Αυτές οι ενότητες ονομάζονται kms-crowddetector, kms-pointerdetector, kms-chroma και kms-platedetector, καθώς μπορεί να επεκταθεί με νέες προσαρμοσμένες λειτουργικές μονάδες.



Σχήμα 3.7: Αρχιτεκτονική μονάδων Kurento. Ο διακομιστής Kurento Media Server μπορεί να επεκταθεί με ενσωματωμένες μονάδες (ανιχνευτής πλήθους, ανιχνευτής ποντικιού, χρώμα, platedetector) και επίσης με άλλες προσαρμόσιμες ενότητες.

3.14.4 RTP Streaming

Ο Kurento εφαρμόζει τον αλγόριθμο Ελέγχου συμφόρησης Google, οπότε είναι σε θέση να παράγει και να αναλύει τόσο τις κεφαλίδες RTP του abs-send-time όσο και τα μηνύματα RTCB REMB. Ενεργοποιείται από τη μετάδοση του χαρακτηριστικού goog-remb στο επίπεδο πολυμέσων στην προσφορά SDP όπως ορίζεται στο RFC 4585.

3.15 Το Kurento σε διαδικτυακή εφαρμογή

Σε υψηλό επίπεδο αφαίρεσης, η εφαρμογή web έχει μια αρχιτεκτονική που αποτελείται από τρία διαφορετικά επίπεδα:

3.15.1 Επίπεδο παρουσίασης (πλευρά πελάτη)

Εδώ είναι ο κώδικας της εφαρμογής που είναι υπεύθυνος για την αλληλεπίδραση με τους τελικούς χρήστες, έτσι ώστε οι πληροφορίες να παρουσιάζονται με ολοκληρωμένο τρόπο. Αυτό αποτελείται από σελίδες HTML με κώδικα Javascript. Είναι υπεύθυνος για την παρουσίαση και την καταγραφή πολυμέσων που βασίζεται στις συγκεκριμένες ενσωματωμένες δυνατότητες του πελάτη.

3.15.2 Λογική εφαρμογών (πλευρά εξυπηρετητή)

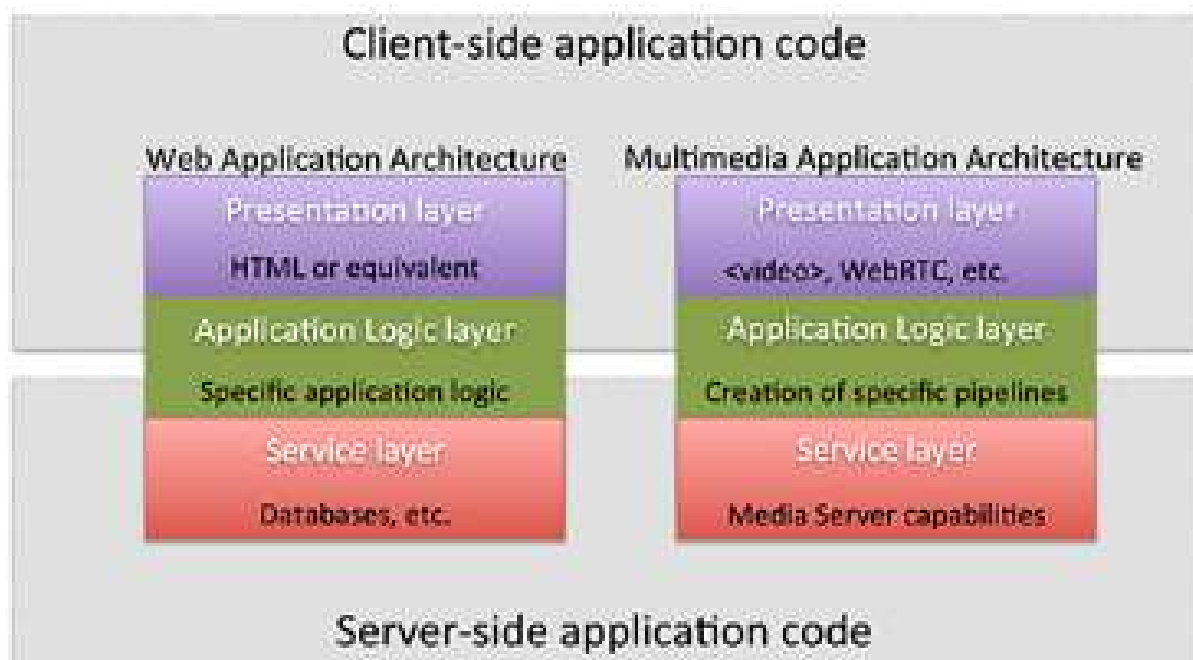
Αυτό το επίπεδο παρέχει τη λογική των πολυμέσων. Είναι υπεύθυνο για την κατασκευή του κατάλληλου αγωγού (με την αλυσίδα των επιθυμητών στοιχείων πολυμέσων), ώστε οι ροές πολυμέσων που εμπλέκονται στην εφαρμογή να διασχίσουν.

3.15.3 Επίπεδο υπηρεσιών (διακομιστής ή διαδίκτυο)

Αυτό το επίπεδο παρέχει δυνατότητες που χρησιμοποιούνται από τη λογική εφαρμογής, όπως βάσεις δεδομένων, επικοινωνίες, ασφάλεια κλπ. Οι υπηρεσίες αυτές μπορούν να φιλοξενοούνται στον ίδιο διακομιστή με τη λογική εφαρμογής ή μπορούν να παρέχονται από εξωτερικούς φορείς. Παρέχει τις υπηρεσίες πολυμέσων που υποστηρίζουν τη λογική της εφαρμογής, όπως εγγραφή πολυμέσων, κρυπτογράφηση πολυμέσων κ.λπ. Υπεύθυνος είναι διακομιστής Kurento Media Server (δηλ. Ο συγκεκριμένος αγωγός πολυμέσων των μέσων επικοινωνίας).

3.16 Αρχιτεκτονική Client-side και Server-side

Οι εφαρμογές Kurento μπορούν να τοποθετήσουν στο επίπεδο παρουσίασης στην πλευρά του πελάτη και το επίπεδο υπηρεσιών, στην πλευρά του διακομιστή. Ωστόσο, η λογική εφαρμογής, και στις δύο περιπτώσεις, μπορεί να βρίσκεται σε οποιαδήποτε από τις πλευρές ή ακόμα και να κατανέμεται μεταξύ τους. Η ιδέα αυτή παρουσιάζεται στην ακόλουθη εικόνα:



Σχήμα 3.8: Αρχιτεκτονική εφαρμογών web και πολυμέσων σε λογικά επίπεδα.

Και οι δύο επιλογές είναι έγκυρες αλλά καθένα από συνεπάγονται διαφορετικά στυλ ανάπτυξης.

3.17 Αρχιτεκτονική εφαρμογών

Το Kurento, όπως και οι περισσότερες τεχνολογίες επικοινωνίας πολυμέσων, είναι χτισμένο χρησιμοποιώντας δύο στάδια.

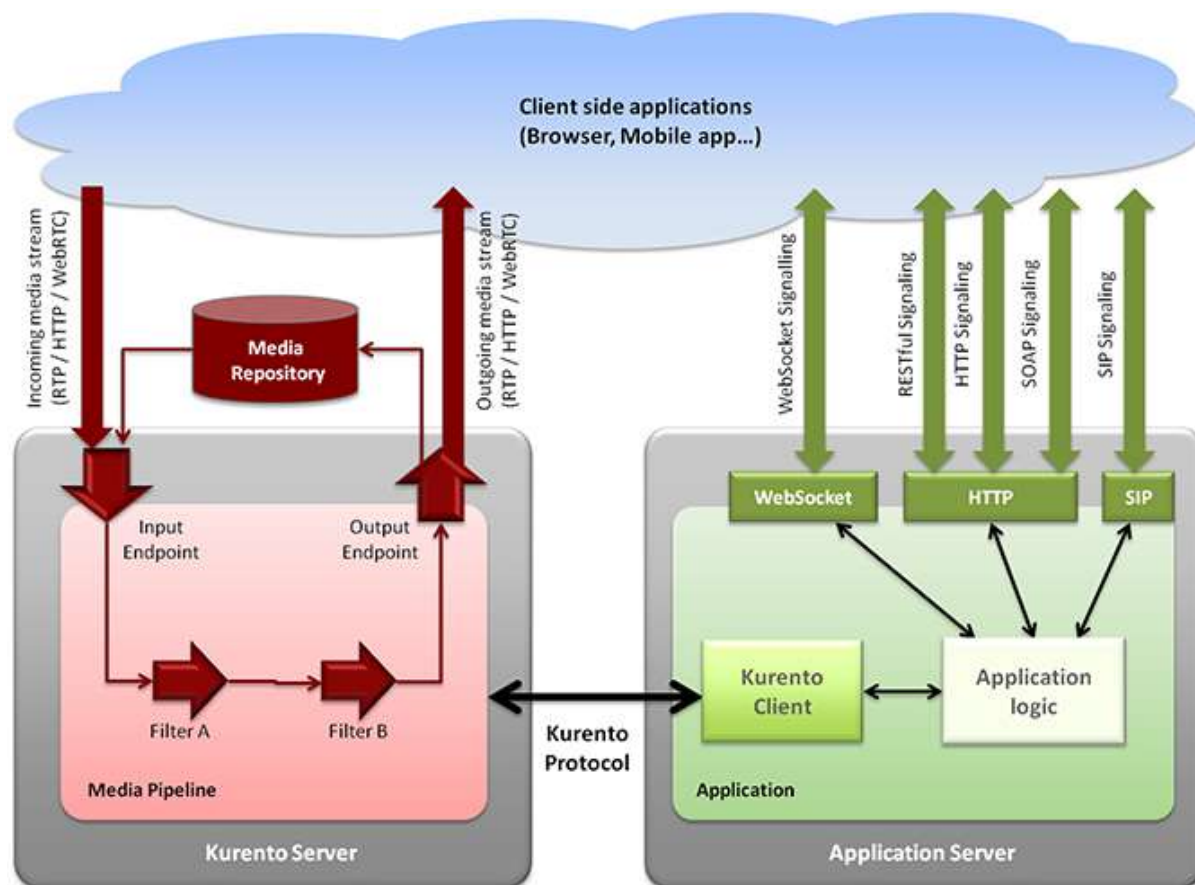
3.17.1 Στάδιο σηματοδότησης (Signaling Plane)

Τα τμήματα του συστήματος που είναι υπεύθυνα για τη διαχείριση των επικοινωνιών, δηλαδή οι μονάδες που παρέχουν λειτουργίες για τη διαπραγμάτευση των μέσων επικοινωνίας, την παραμετροποίηση QoS, την πραγματοποίηση κλήσεων, την εγγραφή χρηστών, και την παρουσίαση χρηστών, θεωρούνται ότι αποτελούν μέρος του σταδίου Σηματοδότησης.

3.17.2 Στάδιο πολυμέσων (Media Plane)

). Λειτουργίες όπως η μεταφορά πολυμέσων, η κωδικοποίηση / αποκωδικοποίηση πολυμέσων και η επεξεργασία πολυμέσων καθιστούν το Media Plane, το οποίο φροντίζει

για το χειρισμό του μέσου. Η διάκριση προέρχεται από τη διαφοροποίηση της τηλεφωνίας μεταξύ του χειρισμού της φωνής και του χειρισμού των μετα-πληροφοριών, όπως ο τόνος και η τιμολόγηση.



Σχήμα 3.9: Εννοιολογική αναπαράσταση της αρχιτεκτονικής του Kurento σε υψηλό επίπεδο

Στη δεξιά πλευρά της εικόνας εμφανίζεται η εφαρμογή, η οποία είναι υπεύθυνη για το επίπεδο σηματοδότησης και περιέχει την επιχειρησιακή λογική και τους συνδέσμους της συγκεκριμένης εφαρμογής πολυμέσων που αναπτύσσεται. Η εφαρμογή μπορεί να χρησιμοποιήσει ώριμες τεχνολογίες όπως διακομιστές HTTP και SIP, υπηρεσίες Web, συνδέσεις βάσεων δεδομένων, υπηρεσίες μηνυμάτων. Αυτό το στάδιο παρέχει πρόσβαση στα πρωτόκολλα σηματοδοσίας πολυμέσων που χρησιμοποιούνται συνήθως από τους τελικούς πελάτες, όπως SIP, RESTful και raw HTTP μορφές, SOAP, RMI, CORBA ή JMS. Αυτά τα πρωτόκολλα σηματοδοσίας χρησιμοποιούνται από την πλευρά του πελάτη των εφαρμογών για να διευθύνουν τη δημιουργία συνόδων πολυμέσων και να διαπραγματευτούν τα επιθυμητά χαρακτηριστικά τους για λογαριασμό τους. Ως εκ τούτου, αυτό είναι το μέρος της αρχιτεκτονικής, που είναι σε επαφή με τους προγραμματιστές

εφαρμογών και για το λόγο αυτό, έχει σχεδιαστεί επιδιώκοντας την απλότητα και την ευελιξία.

Στην αριστερή πλευρά, υπάρχει ο διακομιστής Kurento Media Server, ο οποίος υλοποιεί τις δυνατότητες των πολυμέσων μαζικής επικοινωνίας παρέχοντας πρόσβαση στις δυνατότητες πολυμέσων χαμηλού επιπέδου: μεταφορά μέσων, κωδικοποίηση / αποκωδικοποίηση πολυμέσων, αναδιαμόρφωση πολυμέσων, ανάμειξη πολυμέσων, επεξεργασία πολυμέσων. Ο διακομιστής πρέπει να μπορεί να διαχειρίζεται τις ροές πολυμέσων με ελάχιστη καθυστέρηση και μέγιστη απόδοση. Ως εκ τούτου, ο διακομιστής Kurento Media Server βελτιστοποιείται για την αποδοτικότητα.

3.18 Επικοινωνία πελάτη, διακομιστή και Kurento

Μία εφαρμογή Kurento περιλαμβάνει αλληλεπιδράσεις μεταξύ τριών βασικών ενοτήτων.

3.18.1 Εφαρμογή πελάτη

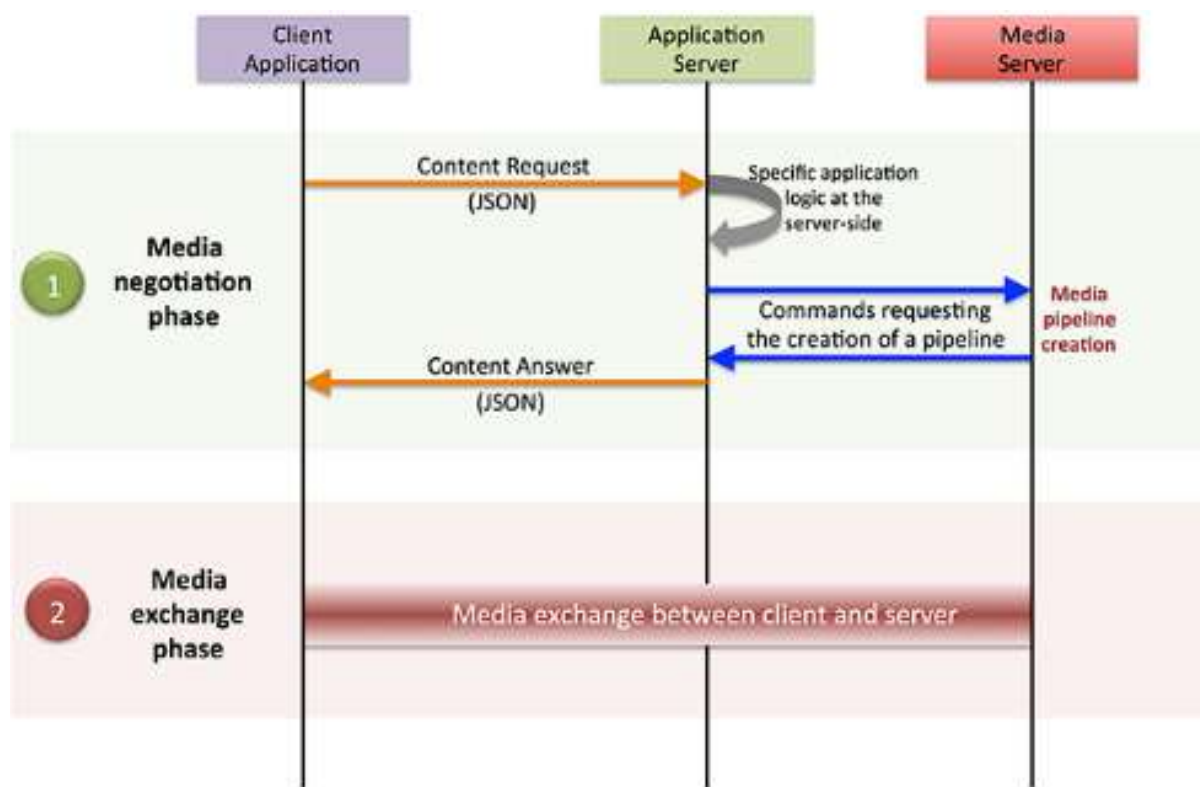
Περιλαμβάνει τις εγγενείς δυνατότητες πολυμέσων της πλατφόρμας πελάτη καθώς και την ιδιαίτερη λογική εφαρμογής της πλευράς του πελάτη. Χρησιμοποιεί Kurento Javascript Client.

3.18.2 Application Server

Συμπεριλαμβάνει ένα διακομιστή εφαρμογών και την λογική της εφαρμογής από την πλευρά του διακομιστή. Χρησιμοποιεί Kurento API, σχεδιασμένο για πλατφόρμες διακομιστών, Kurento Java Client για Java EE και Kurento Javascript Client για Node.js.

3.18.3 Kurento Media Server

Λήψη εντολών για τη δημιουργία συγκεκριμένων δυνατοτήτων πολυμέσων (δηλαδή ειδικών αγωγών προσαρμοσμένων στις ανάγκες της εφαρμογής). Οι αλληλεπιδράσεις που διατηρούνται μεταξύ αυτών των ενοτήτων εξαρτώνται από τις ιδιαιτερότητες κάθε εφαρμογής. Ωστόσο, γενικά, για τις περισσότερες εφαρμογές μπορεί να περιοριστεί στο ακόλουθο εννοιολογικό σχήμα:



Σχήμα 3.10: Δημιουργία αγωγών επικοινωνίας

Κύριες αλληλεπιδράσεις μεταξύ αρχιτεκτονικών ενοτήτων. Αυτά συμβαίνουν σε δύο φάσεις: διαπραγμάτευση και ανταλλαγή μέσων. Τα πορτοκαλί βέλη δείχνουν ανταλλαγές που ανήκουν στο επίπεδο σηματοδότησης, τα μπλε βέλη δείχνουν ανταλλαγές που ανήκουν στο πρωτόκολλο Kurento, τα κόκκινα πλαίσια συνδέονται με τον διακομιστή Kurento Media Server και τα πράσινα πλαίσια με την εφαρμογή.

3.19 Φάση διαπραγμάτευσης των πολυμέσων (σηματοδότηση)

Σε ένα πρώτο στάδιο, ένας πελάτης (ένα πρόγραμμα περιήγησης σε έναν υπολογιστή, μια εφαρμογή για κινητά) εκδίδει ένα μήνυμα στην εφαρμογή που ζητά κάποιο είδος δυνατότητας πολυμέσων. Αυτό το μήνυμα μπορεί να υλοποιηθεί με οποιοδήποτε πρωτόκολλο (HTTP, WebSocket, SIP).

Όταν η εφαρμογή παραλάβει το αίτημα, εάν χρειαστεί, θα εκτελέσει τη συγκεκριμένη λογική εφαρμογής του διακομιστή, η οποία μπορεί να περιλαμβάνει την πιστοποίηση ταυτότητας, την εξουσιοδότηση και τη λογιστική (AAA), την παραγωγή CDR, την

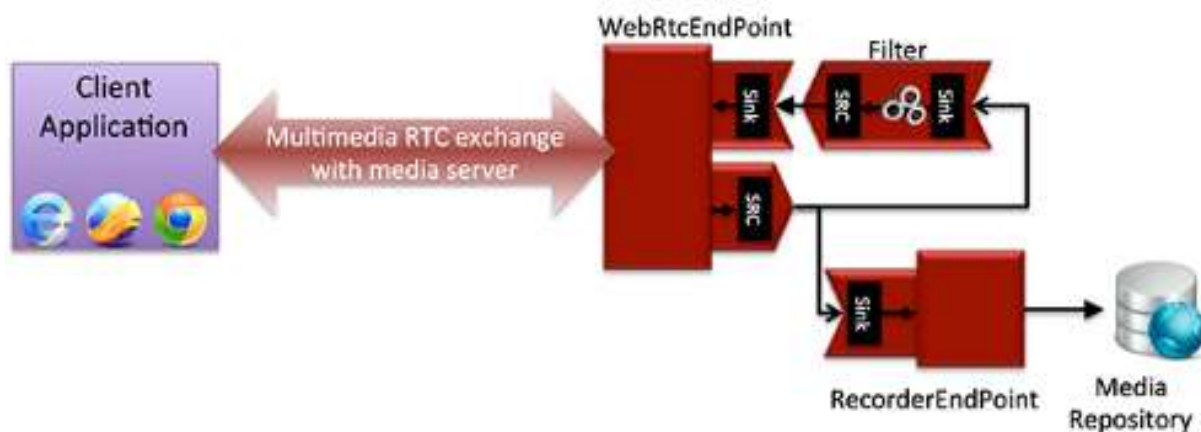
κατανάλωση κάποιου τύπου υπηρεσίας ιστού. Έστερα, η εφαρμογή επεξεργάζεται το αίτημα και σύμφωνα με τις προγραμματισμένες οδηγίες, δίνεται εντολή στον διακομιστή Kurento Media Server να δημιουργήσει τα κατάλληλα στοιχεία μέσων και να τα συνδέσει σε κατάλληλο Media Pipeline. Αφού δημιουργηθεί με επιτυχία ο αγωγός, ο Kurento Media Server ανταποκρίνεται αναλόγως και η εφαρμογή προωθεί την επιτυχή απάντηση στον πελάτη, δείχνοντας πώς και πού μπορεί να επιτευχθεί η υπηρεσία πολυμέσων. Κατά τα προαναφερθέντα βήματα δεν ανταλλάσσονται πραγματικά δεδομένα μέσων. Όλες οι αλληλεπιδράσεις έχουν ως στόχο τη διαπραγμάτευση των τι, πώς, και τον χρόνο, της ανταλλαγής μέσων ενημέρωσης. Για αυτόν τον λόγο ονομάζεται φάση διαπραγμάτευσης. Σαφώς, κατά τη διάρκεια αυτής της φάσης εμπλέκονται μόνο πρωτόκολλα σηματοδότησης.

3.20 Φάση ανταλλαγής πολυμέσων

Μετά το τμήμα σηματοδότησης ξεκινά μια νέα φάση με σκοπό την παραγωγή της πραγματικής ανταλλαγής μέσων. Ο πελάτης απευθύνει αίτημα για τα μέσα στο διακομιστή Kurento Media Server χρησιμοποιώντας τις πληροφορίες που συγκεντρώθηκαν κατά τη διάρκεια της φάσης των διαπραγματεύσεων.

3.21 Εφαρμογές πραγματικού χρόνου

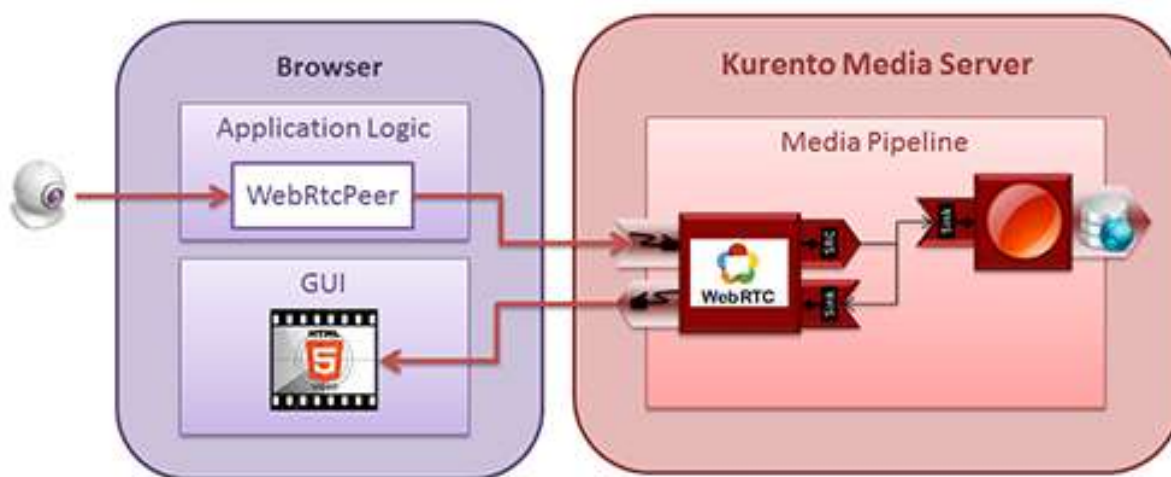
Ο πελάτης επικοινωνεί με τις επιθυμητές δυνατότητες των πολυμέσων του, μέσω μιας διαπραγμάτευσης προσφοράς / απάντησης SDP. Ως εκ τούτου, το Kurento είναι σε θέση να παρουσιάσει το κατάλληλο παράθυρο WebRTC και να του ζητήσει να δημιουργήσει μια απάντηση SDP βάσει των δικών του δυνατοτήτων στην προσφορά SDP. Όταν ληφθεί η απάντηση SDP, δίνεται πίσω στον πελάτη και μπορεί να ξεκινήσει η ανταλλαγή πολυμέσων. Ο προγραμματιστής εφαρμογών είναι σε θέση να δημιουργήσει τον επιθυμητό αγωγό κατά τη διάρκεια της φάσης διαπραγμάτευσης, έτσι ώστε η ροή πολυμέσων σε πραγματικό χρόνο να επεξεργάζεται ανάλογα με τις ανάγκες της εφαρμογής. Στο παρακάτω σχήμα παρουσιάζεται ο αγωγός, όπου ένα στοιχείο φίλτρου πολυμέσων, είναι ικανό να ανιχνεύει και να τροποποιήσει το πολυμέσο.



Σχήμα 3.11: Φίλτρα στην επικοινωνία ανάμεσα από τους χρήστες

3.22 Αγωγοί πολυμέσων

Ένα γράφημα συνδεδεμένων στοιχείων πολυμέσων καλείται Media Pipeline. Συνεπώς, κατά τη δημιουργία ενός αγωγού, προσδιορίζονται οι δυνατότητες - τα Στοιχεία πολυμέσων και η τοπολογία που καθορίζει ποιο φίλτρο πολυμέσων περιέχεται μέσα σε άλλα στοιχεία πολυμέσων.

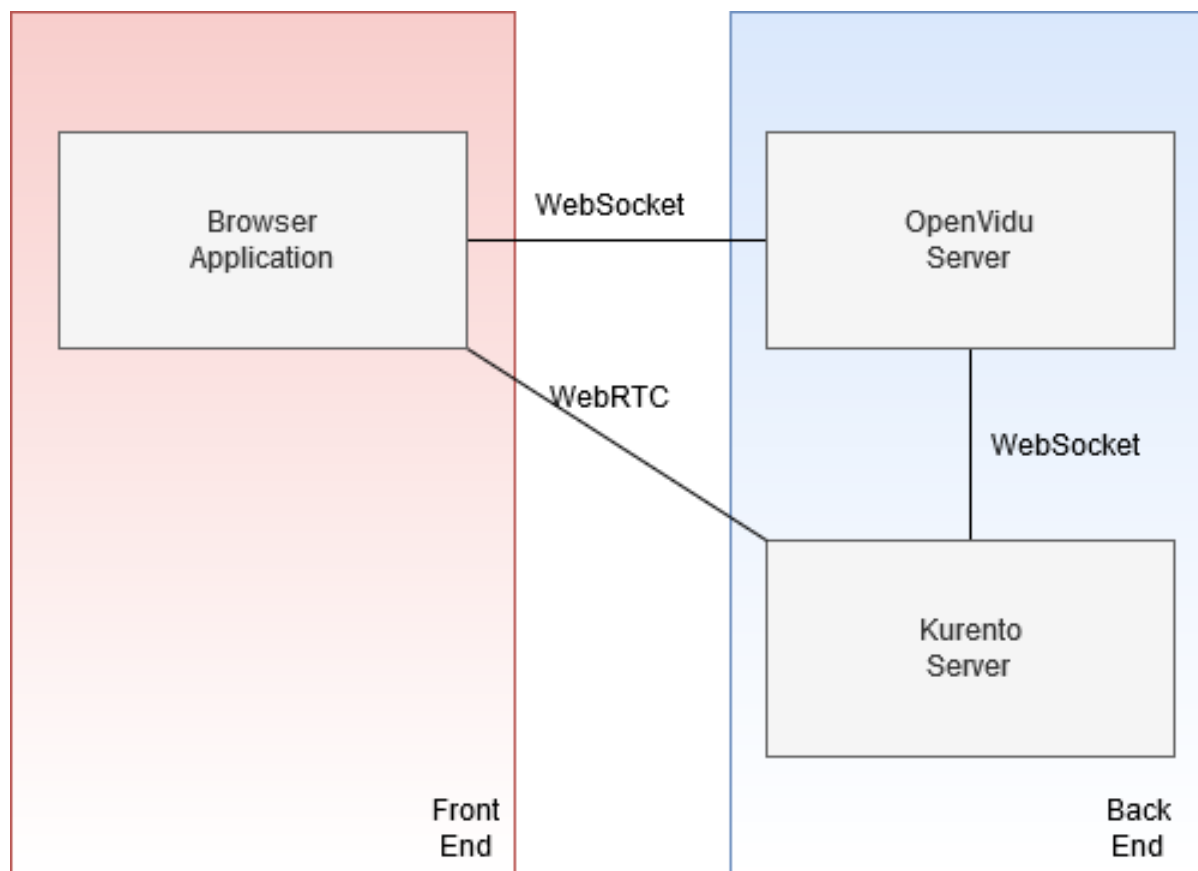


Σχήμα 3.12: Λειτουργία αγωγού πολυμέσων

3.23 OpenVidu

Το OpenVidu είναι μια πλατφόρμα που διευκολύνει την προσθήκη βίντεο κλήσεων σε εφαρμογές διαδικτύου.

3.23.1 Αρχιτεκτονική



Σχήμα 3.13: Εφαρμογές στο εμπρόσθιο κομμάτι και στον εξυπηρετητή

- **OpenVidu Browser:** είναι μια βιβλιοθήκη που χρησιμοποιείται στην πλευρά του πελάτη διαθέσιμο σε Javascript και TypeScript. Επιτρέπει να δημιουργηθούν βίντεο κλήσεις, σύνδεση με χρηστών, η αποστολή και η λήψη βίντεο και ήχου και δεδομένων.
- **OpenVidu Server:** είναι μια εφαρμογή που χειρίζεται τα πράγματα από την πλευρά του διακομιστή. Λαμβάνει τις λειτουργίες από το OpenVidu Browser και κάνει ό, τι είναι απαραίτητο για τη δημιουργία και τη διαχείριση των βίντεο κλήσεων.

3.23.2 Πώς λειτουργεί το OpenVidu

Το WebRTC είναι ο απόλυτος υπεύθυνος για τη μετάδοση των πολυμέσων επικοινωνίας στο OpenVidu. Το Kurento είναι το πλαίσιο στο οποίο είναι κατασκευασμένο το OpenVidu. Το OpenVidu αναδιπλώνει και αποκρύπτει όλες τις λειτουργίες χαμηλού επιπέδου. Ο

κύριος στόχος είναι να προσφέρει ένα απλό, αποτελεσματικό, εύκολο στη χρήση API, ώστε να αποκρύψει από την υλοποίηση τους υποψήφιους για το WebRTC, τους υποψήφιους ICE και το δύσκολο υλικό των διακομιστών πολυμέσων. Χρησιμοποιεί εσωτερικά το Kurento για να δημιουργήσει, να συνδέσει, να τροποποιήσει και να καταστρέψει αγωγούς πολυμέσων, αλλά το OpenVidu απλοποιεί όσο τη διαδικασία, εστιάζοντας την χρήση μόνο των κλήσεων βίντεο. Το OpenVidu είναι μια τεχνολογία WebRTC που επιτρέπει να εφαρμόζονται φίλτρα ήχου και βίντεο σε πραγματικό χρόνο. Το OpenVidu μπορεί να διαμορφωθεί για να καταγράφει τις περιόδους σύνδεσης. Αρχεία OpenVidu:

- OpenVidu-browser.js: είναι η βιβλιοθήκη του OpenVidu για τον περιηγητή ιστού.
- app.js: Το αρχείο App.js είναι κεντρικός πυλώνας της εφαρμογής. Είναι ένα React component που έχει σαν στόχο την δημιουργία Session, την δημιουργία της εφαρμογής στον χρήστη αφού υπάρχει Session και την διαχείριση του δωματίου με τα βίντεο streams σε επίπεδο WebRTC.
- index.html: Αρχείο HTML για τη σελίδα υποδοχής, την συμμετοχή σε ένα νέο δωμάτιο και για την δημιουργία το δωματίου.
- registerServiceWorker.js: Αρχείο OpenVidu για την δημιουργία Web Worker. Χρήσιμο για να σερβίρει παράλληλα με το κύριο νήμα αποθηκευμένα assets της εφαρμογής.
- OnVideo.js: Το αρχείο OnVideo.js περιέχει την κλάση OpenViduVideoComponent. Αυτή είναι ένα component της React και περιέχει το html video tag, που θα προστεθεί κατά την προσθήκη ενός βίντεο stream στην εφαρμογή, για κάποιον publisher/subscriber.
- UserVideoComponent.js: Κύριο αρχείο για το component του Video του χρήστη.
- UserVideo.css: Αρχείο που περιέχει λεπτομέρειες για το πώς θα εμφανιστεί το βίντεο του χρήστη στον χρήστη.
- index.js: Αρχείο εκκίνησης της εφαρμογής.
- index.css: Αρχείο που περιέχει κανόνες για την εμφάνιση της εφαρμογής σε υψηλό επίπεδο.

3.23.3 Επιπλέον λειτουργίες του OpenVidu στην πλατφόρμα

Ο χρήστης εισέρχεται στην εφαρμογή επιλέγοντας ένα όνομα και ένα δωμάτιο. Την διαδικασία αναγνώρισης του χρήστη την αναλαμβάνει ο OpenVidu server.

1. Από τα στοιχεία του χρήστη, δημιουργείται ένα κατακερματισμένο αναγνωριστικό (sessionID). Αυτό μεταφέρεται στον OpenVidu server για να γίνει η ταυτοποίηση και αναγνώριση του χρήστη. Το κάθε αναγνωριστικό είναι μοναδικό και εξασφαλίζει την ταυτότητα του χρήστη στο δωμάτιο.
2. Ο πρώτος χρήστης που εισέρχεται στο δωμάτιο είναι υπεύθυνος για την εκκίνηση δημιουργίας συνεδρίας του δωματίου. Έτσι όλοι οι μετέπειτα χρήστες δεν ξαναδημιουργούν την συνεδρία, αλλά συμμετέχουν.
3. Για κάθε έναν χρήστη, ο εαυτός του θεωρείται publisher και οι υπόλοιποι είναι subscribers. Αυτό γίνεται διότι ο κάθε χρήστης εκπέμπει τον εαυτό του και λαμβάνει την εκπομπή των άλλων χρηστών.
4. Έτσι ο χρήστης, εφόσον εισέλθει στο δωμάτιο βλέπει το όνομά του πάνω από το βίντεο του και καθώς εισέρχονται και άλλοι χρήστες, αντίστοιχα βλέπει το όνομά τους πάνω από το βίντεο τους.
5. Το κάθε δωμάτιο συλλέγει τους χρήστες που το έχουν επιλέξει και αναλαμβάνει την αναμετάδοση βίντεο μεταξύ αυτών. Οι χρήστες του δωματίου συλλέγονται δυναμικά, δηλαδή η κατάσταση του δωματίου είναι πάντα τουλάχιστον open, και έτσι οι χρήστες μπορούν να συμμετάσχουν και να αποχωρήσουν από το δωμάτιο.
6. Έξοδος του χρήστη από την εφαρμογή: Για να βγει ο χρήστης από την συνδιάσκεψη, μπορεί είτε να κλείσει το παράθυρο, είτε να πατήσει το πλήκτρο logout. Σε κάθε περίπτωση το βίντεο που μεταδίδει τερματίζεται, και καλείται η διαδικασία διαγραφής του από το δωμάτιο.
7. Η σύννοδος κάθε χρήστη αποθηκεύεται μέχρι ο χρήστης να την κλείσει ή να κλείσει το πρόγραμμα περιήγησης. Η σύννοδος κάθε χρήστη περιέχει τα εξής χαρακτηριστικά:
 - Το μοναδικό χαρακτηριστικό της συνόδου
 - Το όνομα χρήστη

- Το κύριο βίντεο
- Τα βίντεο που μεταδίδει
- Του συνδρομητές που του μεταδίδουν βίντεο
- Τα προς εγγραφή βίντεο

Η συνεδρία έχει έναν διαχειριστή. Αυτός καλείται με το πρώτο άτομο που εισέρχεται στην συνεδρία.

- Είσοδος στην συνεδρία: Κατά την είσοδο της συνεδρίας, ο ρόλος του διαχειριστή της συνεδρίας είναι να δημιουργήσει και να αρχικοποιήσει έναν κόμβο μεταξύ του χρήστη και του δωματίου, που επιλέγει ο χρήστης να μπει. Επιπλέον ξεκινά η διαδικασία μετάδοσης και λήψης των stream του δωματίου.
- Έξοδος από την συνεδρία: Η έξοδος από την συνεδρία έχει στόχο την έξοδο του χρήστη από ένα δωμάτιο. Έτσι τερματίζεται η κάθε ανταλλαγή πληροφοριών από και προς του άλλους χρήστες του δωματίου καθώς και με τους διακομιστές. Αλλαγή ονόματος χρήστη: Μετά την είσοδο του χρήστη είναι δυνατή η αλλαγή του ονόματός του.

8. Εκκίνηση / Παρακολούθηση συνεδρίας σε ένα δωμάτιο: για κάθε έναν χρήστη, που ήδη μεταδίδει βίντεο στο δωμάτιο και για κάθε έναν καινούργιο χρήστη, που ξεκινά να μεταδίδει, δημιουργείται και ένας αντίστοιχος συνδρομητής. Οι συνδρομητές είναι όλοι οι άλλοι χρήστες, εκ των οποίων ένας χρήστης, λαμβάνει ροές βίντεο. Οι συνδρομητές είναι αποθηκευμένοι σε μια λίστα για την διαχείρισή τους. Όταν ένας συνδρομητής φύγει από το δωμάτιο, το βίντεο που εκπέμπει σταματά και διαγράφεται από την λίστα συνδρομητών. Ο χρήστης δημιουργεί ένα τεκμήριο (token) που συνδέει το όνομά του με την συνεδρία. Μετά ξεκινά την μετάδοση του δικού του βίντεο στην συνεδρία. Αυτό το βίντεο ορίζεται αρχικά και ως κύριο βίντεο για τον χρήστη καθώς και ως βίντεο εκπομπής. Όλες οι συνεδρίες μπορούν να καταγραφούν τοπικά από τον χρήστη.
9. Αποχώρηση από την συνομιλία: για να αποχωρήσει ο χρήστης από την συνεδρία, είτε από ομαλή διαδικασία αποχώρησης, είτε από ομαλό κλείσιμο του προγράμματος, καλείται η διαδικασία αποχώρησης. Αυτή αποθηκεύει τα βίντεο που εγγράφονται

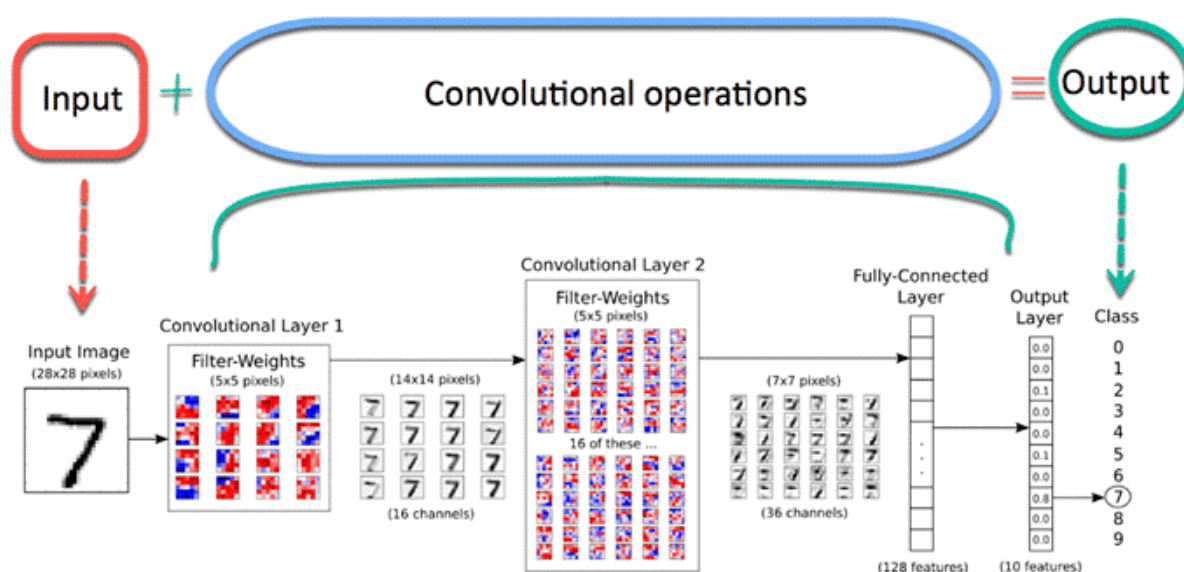
και καλεί την διαδικασία αποσύνδεσης από τους άλλους χρήστες. Σε περίπτωση απλής αποχώρησης από την συνεδρία, το στοιχείο App και η συνεδρία, επανέρχονται στην αρχική τους κατάσταση.

10. Διαγραφή Συνδρομητή: Η διαγραφή συνδρομητή, καλείται όταν ένας συνδρομητής αποχωρεί από το δωμάτιο. Τότε οι υπόλοιποι χρήστες πρέπει να τον αφαιρέσουν από την λίστα των συνδρομητών τους.
11. Κατασκευή τεκμηρίου / συνεδρίας: Η κατασκευή του τεκμηρίου όπως και η κατασκευή της συνεδρίας γίνονται από τον OpenVidu εξυπηρετητή. Αρχικά ο OpenVidu εξυπηρετητής κρατάει την συνεδρία του χρήστη, όταν αυτός εισέρχεται στην εφαρμογή και έπειτα την χρησιμοποιεί για να φτιάξει το τεκμήριο.

3.24 Μηχανική Μάθηση

3.24.1 Αρχιτεκτονική ενός Συνελικτικού νευρωνικού δικτύου

Μια εικόνα εισόδου υποβάλλεται σε επεξεργασία κατά τη διάρκεια της φάσης συνέλιξης και αργότερα αποδίδεται μια ετικέτα. Αυτό ονομάζεται εικόνα εισόδου. Στη συνέχεια, η εικόνα εισόδου περνάει από άπειρο αριθμό βημάτων. Αυτό είναι το συνελικτικό μέρος του δικτύου. Τέλος, το νευρικό δίκτυο μπορεί να προβλέψει το ψηφίο στην εικόνα. Μια τυπική αρχιτεκτονική convnet μπορεί να συνοψιστεί στην παρακάτω εικόνα.



Σχήμα 3.14: Λειτουργία συνελικτικού δικτύου σε μια εικόνα

Μια εικόνα αποτελείται από μια σειρά από εικονοστοιχεία με ύψος και πλάτος. Μια εικόνα σε κλίμακα του γκρι έχει μόνο ένα κανάλι ενώ η έγχρωμη εικόνα έχει τρία κανάλια (κόκκινο, πράσινο και μπλε). Ένα κανάλι στοιβάζεται το ένα πάνω στο άλλο. Στα πιο πολλά σενάρια, θα χρησιμοποιείται μια εικόνα κλίμακας του γκριζου με ένα μόνο κανάλι. Κάθε εικονοστοιχείο έχει μια τιμή από 0 έως 255 για να αντικατοπτρίζει την ένταση του χρώματος. Για παράδειγμα, ένα εικονοστοιχείο ίσο με το 0 θα εμφανίσει ένα λευκό χρώμα, ενώ το εικονοστοιχείο με τιμή κοντά στο 255 θα είναι πιο σκούρο.

3.24.2 Συνελικτική Λειτουργία

Το πιο κρίσιμο συστατικό του μοντέλου είναι το φίλτρο συνέλιξης. Αυτό το μέρος στοχεύει στη μείωση του μεγέθους της εικόνας για ταχύτερους υπολογισμούς των βαρών και στη βελτίωση της γενίκευσης. Κατά τη διάρκεια του συνελικτικού τμήματος, το δίκτυο διατηρεί τα βασικά χαρακτηριστικά της εικόνας και αποκλείει τον άσχετο θόρυβο. Για παράδειγμα, το μοντέλο μαθαίνει πώς να αναγνωρίζει έναν ελέφαντα από μια εικόνα με ένα βουνό στο βάθος. Το μοντέλο ενός γενικό νευρωνικού δικτύου, θα αποδώσει βάρος σε όλα τα εικονοστοιχεία, συμπεριλαμβανομένων εκείνων από το βουνό, που δεν είναι απαραίτητα και μπορεί να παραπλανήσει το δίκτυο. Αντίθετα, ένα συνελικτικό νευρωνικό δίκτυο θα χρησιμοποιήσει μια μαθηματική τεχνική για την εξαγωγή μόνο των πιο σχετικών εικονοστοιχείων. Αυτή η μαθηματική λειτουργία ονομάζεται συνέλιξη. Αυτή η τεχνική επιτρέπει στο δίκτυο να μαθαίνει όλο και πιο σύνθετα χαρακτηριστικά σε κάθε επίπεδο. Η συνέλιξη χωρίζει τη τον πίνακα εικονοστοιχείων σε μικρά κομμάτια για να μάθει τα περισσότερα βασικά στοιχεία μέσα σε κάθε κομμάτι.

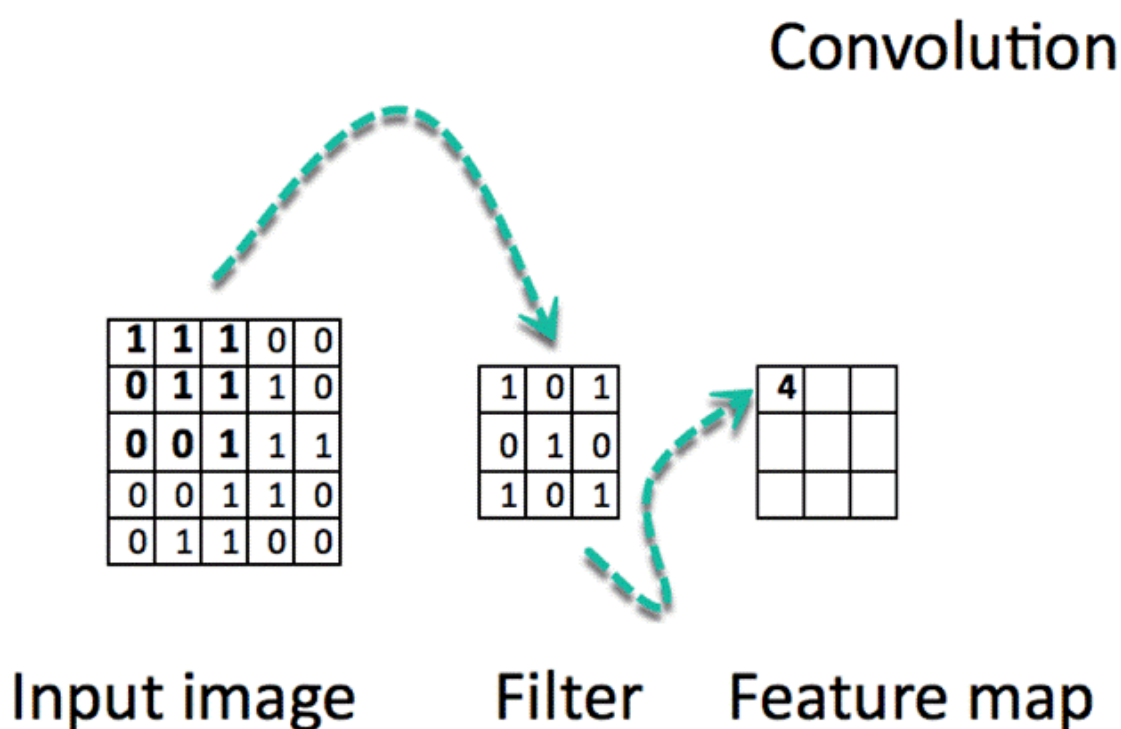
3.24.3 Στατιστικά των Convents και συνέλιξη

Το Convents έχει να κάνει με την κατασκευή του μοντέλου ενός συνελικτικού νευρωνικού δικτύου από το μηδέν. Η αρχιτεκτονική του δικτύου περιέχει έναν συνδυασμό των ακόλουθων βημάτων:

- Συνέλιξη
- Μη Γραμμικότητα (ReLU)
- Συγκέντρωση ή υποδειγματοληψία

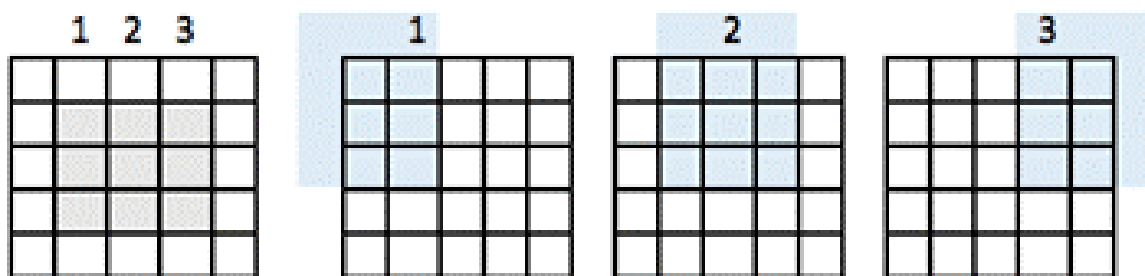
- Ταξινόμηση (πλήρως συνδεδεμένο επίπεδο)

Ο σκοπός της συνέλιξης είναι να εξάγει τοπικά τα χαρακτηριστικά του αντικειμένου πάνω στην εικόνα. Αυτό σημαίνει ότι το δίκτυο θα μάθει συγκεκριμένα μοτίβα μέσα στην εικόνα και θα μπορεί να τα αναγνωρίσει παντού στην εικόνα. Η συνέλιξη είναι ένας πολλαπλασιασμός με στοιχεία. Ο υπολογιστής θα σαρώσει ένα μέρος της εικόνας και θα το πολλαπλασιάσει με ένα φίλτρο. Η έξοδος του πολλαπλασιασμού των στοιχείων ονομάζεται πίνακας χαρακτηριστικών. Αυτό το βήμα επαναλαμβάνεται μέχρι να σαρωθεί ολόκληρη η εικόνα. Με το πέρας της πράξης, το μέγεθος της εικόνας μειώνεται.



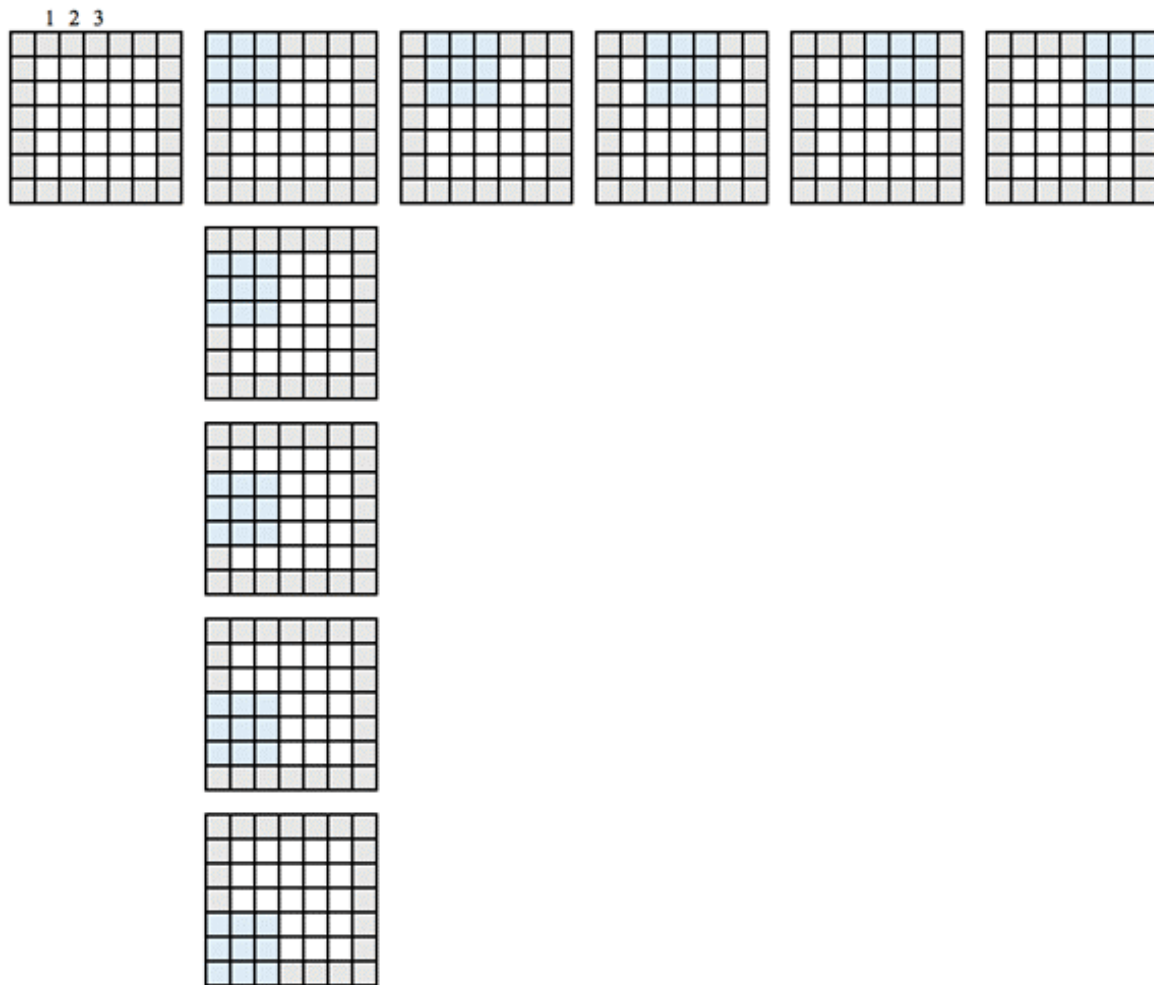
Σχήμα 3.15: Λειτουργία συνέλιξης

Στην συνελικτική φάση εφαρμόζεται το φίλτρο σε μια μικρή σειρά εικονοστοιχείων μέσα στην εικόνα. Το φίλτρο κινείται κατά μήκος της εικόνας εισόδου με ένα γενικό σχήμα, όπως 3x3 ή 5x5. Αυτό σημαίνει ότι το δίκτυο μεταφέρει τους πίνακες σε όλη την εικόνα εισόδου και υπολογίζει την συνέλιξη.



Σχήμα 3.16: Αποτέλεσμα συνέλιξης: το μέγεθος του πίνακα είναι 3×3 και ο πίνακας εξόδου είναι το αποτέλεσμα της λειτουργίας μεταξύ του πίνακα της εικόνας και του φίλτρου

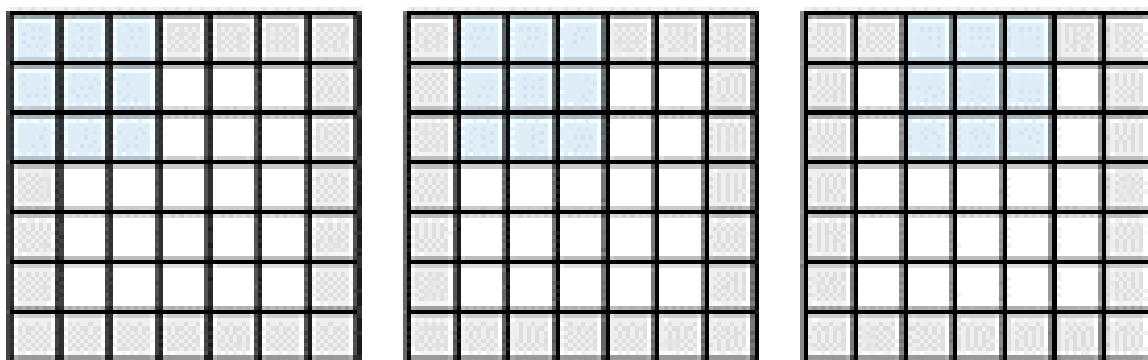
Έστω μια εικόνα έχει έναν χάρτη χαρακτηριστικών 5×5 και ένα φίλτρο 3×3 . Υπάρχει μόνο ένα παράθυρο στο κέντρο όπου το φίλτρο μπορεί να προβάλλει ένα πλέγμα 3×3 . Ο χάρτης χαρακτηριστικών εξόδου θα συρρικνωθεί κατά δύο πλακίδια μαζί με μια διάσταση 3×3 . Αυτό ονομάζεται γειτονικό φαινόμενο. Για να υπάρχει η ίδια διάσταση εξόδου με τη διάσταση εισαγωγής, πρέπει να προστεθούν παραπάνω στοιχεία (Padding). Το Padding αποτελείται από την προσθήκη του σωστού αριθμού σειρών και στηλών σε κάθε πλευρά του πίνακα. Έτσι θα επιτρέψει την συνέλιξη να κεντράρει κάθε θέση εισόδου. Στην παρακάτω εικόνα, ο πίνακας εισόδου / εξόδου έχει την ίδια διάσταση 5×5 .



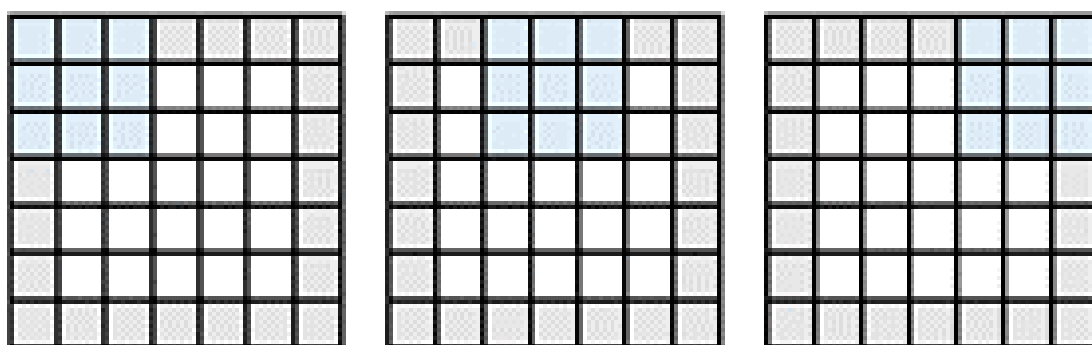
Σχήμα 3.17: Εφαρμογή της συνέλιξης σε όλα τα pixels

Όταν ορίζεται το δίκτυο, τα χαρακτηριστικά της συνέλιξης καθορίζονται από τρεις παραμέτρους:

- **Βάθος** Καθορίζει τον αριθμό των φίλτρων που εφαρμόζονται κατά τη διάρκεια της συνέλιξης. Το βάθος ένα, σημαίνει ότι χρησιμοποιείται μόνο ένα φίλτρο. Στην πλειονότητα των περιπτώσεων, υπάρχουν περισσότερα από ένα φίλτρα. Η παρακάτω εικόνα δείχνει τις λειτουργίες που γίνονται σε μια κατάσταση με τρία φίλτρα.
- **Βηματισμός** Ορίζει τον αριθμό του άλματος "pixel" ανάμεσα σε δύο χωρίσματα. Εάν το βήμα είναι ίσο με το 1, τα παράθυρα θα κινούνται με ένα διάκενο ενός pixel. Εάν το βήμα είναι δύο, τα παράθυρα θα μεταβούν κατά 2 εικονοστοιχεία. Αν αυξηθεί το βήμα, θα είναι μικρότεροι οι χάρτες χαρακτηριστικών.



Σχήμα 3.18: Συνέλιξη με βηματισμό 1



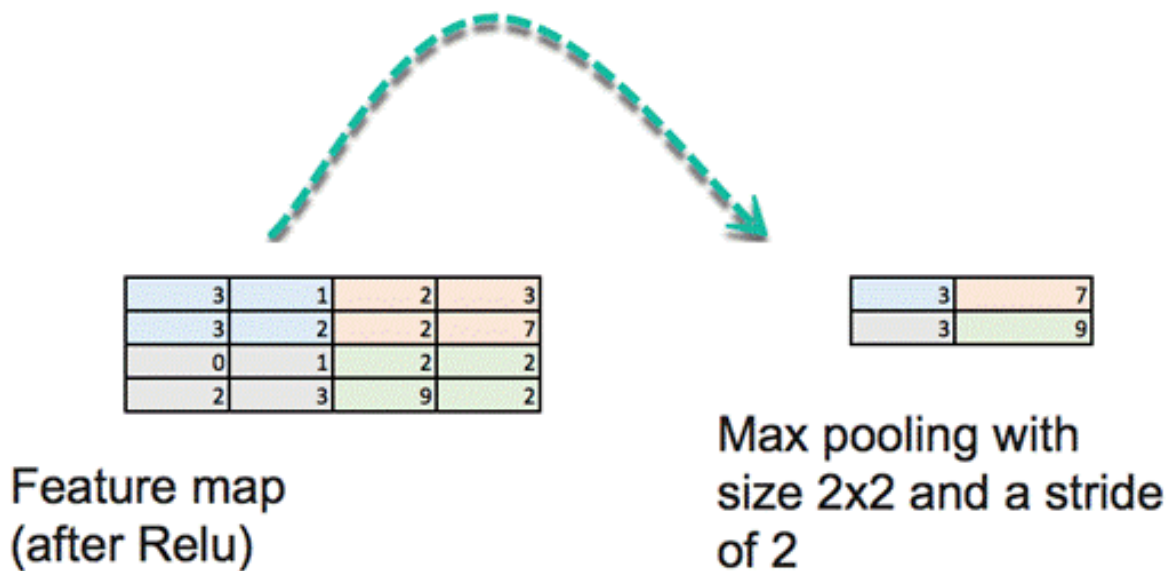
Σχήμα 3.19: Συνέλιξη με βηματισμό 2

- Zero-padding Ονομάζεται η διαδικασία συμπλήρωσης αριθμού γραμμών και στηλών σε κάθε πλευρά των χαρτών χαρακτηριστικών εισόδου για να έχουν ίδια διάσταση με την είσοδο
- Μη Γραμμικότητα (ReLU) Στο τέλος της λειτουργίας συνέλιξης, η έξοδος υπόκειται σε μια συνάρτηση ενεργοποίησης για να επιτρέψει τη μη γραμμικότητα. Η συνήθης λειτουργία ενεργοποίησης για το convnet είναι το Relu. Το σύνολο των εικονοστοιχείων με αρνητική τιμή θα αντικατασταθεί από το μηδέν.

3.24.4 Λειτουργία μέγιστης συγκέντρωσης (Max-pooling operation)

Ο σκοπός της λειτουργία μέγιστης συγκέντρωσης είναι να μειωθεί η διάσταση της εικόνας εισόδου. Τα βήματα γίνονται για να μειωθεί η υπολογιστική πολυπλοκότητα. Μειώνοντας τη διάσταση, το δίκτυο έχει λιγότερα βάρη για τον υπολογισμό, επομένως αποφορτίζεται.

Σε αυτό το στάδιο, πρέπει να οριστεί το μέγεθος και το βήμα. Ένας τυπικός τρόπος συγκέντρωσης της εικόνας εισόδου είναι να χρησιμοποιηθεί η μέγιστη τιμή του χάρτη χαρακτηριστικών. Στο παράδειγμα, η συγκέντρωση θα προβάλλει τέσσερις υποκατηγορίες του χάρτη χαρακτηριστικών 4x4 και θα επιστρέψει τη μέγιστη τιμή. Η συγκέντρωση παίρνει τη μέγιστη τιμή ενός πίνακα 2x2 και στη συνέχεια να μετακινούνται αυτά τα παράθυρα κατά δύο εικονοστοιχεία. Για παράδειγμα, ο πρώτος υπό-πίνακας είναι [3,1,3,2] και η συγκέντρωση θα επιστρέψει το μέγιστο, το οποίο είναι 3.



Σχήμα 3.20: Συγκέντρωση με βάση το μέγιστο στοιχείο

Υπάρχουν και άλλες λειτουργίες συγκέντρωσης όπως ο μέσος όρος. Αυτή η λειτουργία μειώνει δραματικά το μέγεθος του χάρτη χαρακτηριστικών. Πλήρως συνδεδεμένα επίπεδα: Το τελευταίο βήμα συνίσταται στην οικοδόμηση ενός παραδοσιακού τεχνητού νευρικού δικτύου. Συνδέονται όλοι οι νευρώνες από το προηγούμενο επίπεδο στο επόμενο επίπεδο. Χρησιμοποιείται μια συνάρτηση ενεργοποίησης όπως η softmax για να ταξινομηθεί η εικόνα εισόδου.

3.24.5 Πολλαπλά συνελκτικά δίκτυα με Keras

Τα πέντε βήματα στον κύκλο ζωής είναι τα εξής:

1. Ορισμός του μοντέλου.
2. Μεταγλώττιση του μοντέλου.

3. Τοποθέτηση κατάλληλου πλαισίου.
4. Αξιολόγηση του μοντέλου.
5. Προβλέψεις.

3.24.6 Ορισμός του μοντέλου

Ο ορισμός του μοντέλου απαιτεί πρώτα την επιλογή του τύπο μοντέλου που χρειάζεται και στη συνέχεια να η επιλογή της αρχιτεκτονικής ή της τοπολογίας του δικτύου. Δηλαδή ο καθορισμός των επιπέδων του νευρωνικού δικτύου, η σύνδεση κάθε επιπέδου με κόμβους, μια activation function και την σύνδεση των επιπέδων σε ένα συνεκτικό μοντέλο. Τα μοντέλα μπορούν να οριστούν με ακολουθιακό ή με συναρτησιακό τρόπο.

3.24.7 Ακολουθιακό Μοντέλο

Το διαδοχικό API επιτρέπει την δημιουργία μοντέλων στρώμα-στρώμα και είναι επαρκές για τα περισσότερα προβλήματα. Είναι περιορισμένο στο ότι δεν επιτρέπει τη δημιουργία μοντέλων που μοιράζονται στρώματα ή έχουν πολλαπλές εισόδους ή εξόδους.

3.24.8 Συναρτησιακό Μοντέλο

Το συναρτησιακό API επιτρέπει την δημιουργία μοντέλων που έχουν πολύ μεγαλύτερη ευελιξία, καθώς μπορούν εύκολα να καθοριστούν μοντέλα όπου τα επίπεδα συνδέονται με περισσότερο από το προηγούμενο και το επόμενο επίπεδο, με την σύνδεση σε οποιοδήποτε άλλο στρώμα. Ως αποτέλεσμα, είναι δυνατή η δημιουργία πολύπλοκων δικτύων. Στην συγκεκριμένη περίπτωση ακολουθείται ο ακολουθιακός τρόπος, για την δημιουργία διδιάστατων στρώσεων με activation function την “Relu”. Μετά την δημιουργία τους, τα στρώματα συγκεντρώνονται και φιλτράρονται μέσα από ένα pooling layer, για να μειωθεί ο αριθμός των δεδομένων για ανάλυση. Έπειτα, τα δεδομένα κανονικοποιούνται σε μονοδιάστατο πίνακα και στη συνέχεια προστίθενται σε ένα πλήρες συνδεδεμένο στρώμα.

3.24.9 Μεταγλώττιση του μοντέλου

Για μεταγλώττιση του μοντέλου χρειάζεται πρώτα η επιλογή μιας loss function, όπως της mean squared error ή cross-entropy. Επιπλέον, απαιτείται ένας αλγόριθμος για

βελτιστοποίηση, όπως ο stochastic gradient descent ή μια σύγχρονη παραλλαγή, όπως ο αλγόριθμος Adam. Έτσι, για την εκπαίδευση του δικτύου χρησιμοποιήθηκε ο αλγόριθμος Adam, και για την loss function η cross-entropy.

3.24.10 Τοποθέτηση κατάλληλου πλαισίου

Η τοποθέτηση του μοντέλου σε ένα πλαίσιο, απαιτεί πρώτα τη διαμόρφωση της εκπαίδευσης, τον αριθμό των epochs (βρόχους μέσα στο σύνολο δεδομένων εκπαίδευσης) και το μέγεθος του batch (αριθμός δειγμάτων σε ένα epoch που χρησιμοποιείται για την εκτίμηση του σφάλματος μοντέλου). Ρυθμίζοντας αυτές τις μεταβλητές γίνεται βελτιστοποίηση του μοντέλου έτσι ώστε να μην υπάρχει το φαινόμενο της υπέρ-εκπαίδευσης, επάνω στα δεδομένα εκπαίδευσης. Η εκπαίδευση εφαρμόζεται με βάση τον Adam, για να ελαχιστοποιηθούν τα αποτελέσματα της συνάρτησης απώλειας (cross-entropy) και ενημερώνει το μοντέλο χρησιμοποιώντας backpropagation του αλγορίθμου σφάλματος.

3.24.11 Δημιουργία πρόβλεψης

Η πρόβλεψη είναι το τελικό βήμα στον κύκλο ζωής. Εδώ χρειάζονται δεδομένα που δεν έχουν συμπεριληφθεί στην εκπαίδευση του μοντέλου, αλλά έχουν την ίδια σύσταση και είναι κατηγοριοποιημένα. Έτσι φαίνονται τα αποτελέσματα της εκπαίδευσης του μοντέλου, σε διαφορετικά δεδομένα.

3.25 Confusion Matrix (Πίνακας Σύγχυσης)

Παρακάτω είναι ο πίνακας σύγχυσης του μοντέλου, με βάση τα δεδομένα του Kaggle.

Κατηγορία\Πρόβλεψη	Εκνευρισμός	Αηδία	Φόβος	Χαρά	Λύπη	Έκπληξη	Ουδέτερο
Εκνευρισμός	214	9	53	30	67	8	86
Αηδία	10	24	9	2	6	0	5
Φόβος	45	2	208	29	89	45	78
Χαρά	24	0	40	696	37	18	80
Λύπη	65	3	83	56	285	10	151
Έκπληξη	7	1	42	27	9	303	26
Ουδέτερο	45	2	68	65	88	8	331

Πίνακας 3.1: Πίνακας Σύγχυσης Keras

Το ποσοστό επιτυχίας είναι 57%. Με την βοήθεια του Auto-Keras όμως ανεβαίνει στο 66%.

Κατηγορία\Πρόβλεψη	Εκνευρισμός	Αηδία	Φόβος	Χαρά	Λύπη	Έκπληξη	Ουδέτερο	Ανάκληση
Εκνευρισμός	233	9	49	23	90	11	52	50%
Αηδία	12	31	1	2	5	0	5	55%
Φόβος	30	3	189	21	130	41	82	38%
Χαρά	7	1	11	778	22	17	59	87%
Λύπη	35	4	51	24	391	9	139	60%
Έκπληξη	8	1	20	18	13	333	22	80%
Ουδέτερο	13	1	29	49	90	3	422	70%
Ευστοχία	69%	62%	54%	85%	53%	80%	54%	acc=66%

Πίνακας 3.2: Πίνακας Σύγχυσης Auto-Keras

3.25.1 Πολλαπλά συνελικτικά δίκτυα με OpenVino

Πλήρες συνελικτικό δίκτυο για αναγνώριση πέντε συναισθημάτων («ουδέτερο», «χαρούμενο», «λυπημένο», «έκπληξη», «θυμό») Για την αξιολόγηση των μετρήσεων, χρησιμοποιείται το τμήμα επικύρωσης του συνόλου δεδομένων AffectNet. Επιλέγεται ένα υποσύνολο με μόνο τις εικόνες που περιέχουν τα πέντε προαναφερθέντα συναισθήματα. Το συνολικό ποσό των εικόνων που χρησιμοποιούνται στην επικύρωση είναι 2.500. Το μοντέλο είναι σχεδιασμένο σε Caffe και έχει τα εξής χαρακτηριστικά:

Metric	Value	Τα αρχεία που χρησιμοποιούνται είναι τα "/opt/intel/openvino_2019.1.144/inference_engine/samples/python_samples/classification_sample/classification_sample.xml" και "/opt/intel/openvino_2019.1.144/inference_engine/samples/python_samples/classification_sample/int8/emotion_recognition-retail-0003.xml"]
Προσανατολισμού προσώπου	Μετωπικός	
Ελάχιστο μέγεθος εικόνας	64x64 pixels	
GFlops	0.126	
MParams	2.483	
Source framework	Caffe	

Πίνακας 3.3: Λεπτομέρειες μοντέλου OpenVino

4 Τεχνολογίες

4.1 Βιβλιοθήκες μηχανικής μάθησης

Για μηχανική μάθηση σε εικόνες χρειάζεται αρχικά μια τεχνολογία μηχανικής όρασης. Η μηχανική όραση έχει ως σκοπό την ανάλυση μιας εικόνας μέσω αλγορίθμων μηχανικής μάθησης.

4.1.1 TensorFlow

Το TensorFlow είναι μια δωρεάν συμβολική βιβλιοθήκη μαθηματικών, ανοιχτού κώδικα και χρησιμοποιείται για εφαρμογές μηχανικής μάθησης. Είναι υλοποιημένη από την Google και χρησιμοποιείται για εφαρμογές έρευνας και εμπορικές εφαρμογές. Το TensorFlow υποστηρίζεται σε πολλές γλώσσες προγραμματισμού:

- Python
- C/C++
- Java
- Go
- Javascript
- R
- Julia
- Swift

Το κύριο εργαλείο λογισμικού της βαθιάς μάθησης είναι το TensorFlow. Είναι μια βιβλιοθήκη τεχνητής νοημοσύνης ανοιχτού κώδικα, χρησιμοποιώντας γραφήματα ροής δεδομένων για την κατασκευή μοντέλων. Επιτρέπει την δημιουργία μεγάλης κλίμακας νευρωνικών δικτύων με πολλά στρώματα. Το TensorFlow χρησιμοποιείται κυρίως για: Εφαρμογές βασισμένες σε κείμενο: Δημοφιλείς χρήσεις του TensorFlow είναι οι εφαρμογές που βασίζονται σε κείμενα όπως η συναισθηματική ανάλυση, η ανίχνευση και η ανίχνευση απάτης. Αναγνώριση εικόνων: Η αναγνώριση προσώπου, η αναζήτηση εικόνων, η ανίχνευση κίνησης, η μηχανική μάθηση και η ομαδοποίηση

φωτογραφιών, χρησιμοποιούνται στις βιομηχανίες αυτοκινήτων, αερομεταφορών και υγειονομικής περίθαλψης. Η αναγνώριση εικόνας αποσκοπεί στην αναγνώριση ατόμων και αντικειμένων στις εικόνες, καθώς και στην κατανόηση του περιεχομένου του πλαισίου. Οι αλγόριθμοι αναγνώρισης αντικειμένων TensorFlow ταξινομούν και εντοπίζουν αυθαίρετα αντικείμενα μέσα σε μεγαλύτερες εικόνες, για τον εντοπισμό σχημάτων για σκοπούς μοντελοποίησης (3D διαστημική κατασκευή από 2D εικόνες) και από κοινωνικά δίκτυα για την ετικέτα φωτογραφιών (Deep Face του Facebook). Αναλύοντας χιλιάδες φωτογραφίες για παράδειγμα, η τεχνολογία μπορεί να μάθει να αναγνωρίζει ένα αντικείμενο που δεν έχει ξαναδεί ποτέ. Η αναγνώριση εικόνας επεκτείνεται και στην Βιομηχανία Υγείας, όπου οι αλγόριθμοι TensorFlow μπορούν να επεξεργάζονται περισσότερες πληροφορίες και να εντοπίζουν περισσότερα μοτίβα και έτσι να εντοπίσουν περισσότερες ασθένειες από τους ανθρώπους. Χρονοσειρές: Οι αλγόριθμοι TensorFlow Time Series χρησιμοποιούνται για την ανάλυση δεδομένων χρονοσειρών προκειμένου να εξαχθούν σημαντικά στατιστικά στοιχεία. Επιτρέπουν την πρόβλεψη μη συγκεκριμένων χρονικών περιόδων και της δημιουργίας εναλλακτικών εκδόσεων των χρονοσειρών. Η συνηθέστερη περίπτωση χρήσης για τη Σειρά Χρόνου είναι η σύσταση, δηλαδή η ανάλυση της δραστηριότητα των πελατών και η σύγκριση με εκατομμύρια άλλων χρηστών για να καθορίσουν τι μπορεί να αγοράσει ή να παρακολουθήσει ο πελάτης.

Οι βιβλιοθήκες του TensorFlow που χρησιμοποιήθηκαν είναι:

- `npm/@TensorFlow/tfjs`: Είναι η βιβλιοθήκη για μηχανική μάθηση της google, TensorFlow. Η συγκεκριμένη έκδοση δουλεύει για Javascript, δίνοντας την δυνατότητα για μηχανική μάθηση στον περιηγητή του χρήστη.
- `npm/@TensorFlow-models/mobilenet`: Είναι το μη εκπαιδευμένο μοντέλο για μηχανική μάθηση πάνω στο οποίο έγινε εκπαίδευση για την αναγνώριση συναισθημάτων βασισμένη σε πρόσωπα.
- `npm/@TensorFlow-models/knn-classifier`: Είναι το μοντέλο στο οποίο αποθηκεύεται και φορτώνεται στην μνήμη το προ-εκπαιδευμένο μοντέλο Mobilenet. Πρακτικά είναι ένας τρόπος να αποθηκεύει και να επαναχρησιμοποιεί το TensorFlow εκπαιδευμένα μοντέλα.

4.1.2 Caffe

Το Caffe είναι μια δωρεάν βιβλιοθήκη ανοιχτού κώδικα, για βαθιά μάθηση που γίνεται με γνώμονα την έκφραση, την ταχύτητα και τη διαμόρφωση. Αναπτύσσεται από την Berkeley AI Research (BAIR) και από κοινοτική συνεισφορά. Αυτό που διαφοροποιεί το Caffe από τα υπόλοιπα μοντέλα μηχανικής μάθησης είναι η ταχύτητα των μοντέλων του. Το Caffe έχει επεκταθεί σε Caffe2 που με βάση αυτό η Facebook έχει δημιουργήσει το PyTorch. Το Caffe υποστηρίζεται στις γλώσσες προγραμματισμού:

- Python
- MATLAB
- C++

4.1.3 OpenVINO

Το OpenVINO (Open Visual Inference Neural Network Optimization) toolkit, είναι ένα εργαλείο της Intel και παρέχει βελτιωμένη απόδοση νευρωνικών δικτύων σε επεξεργαστές Intel®, επιτρέποντας οικονομικές και αποδοτικές εφαρμογές μηχανικής όρασης σε πραγματικό χρόνο. Υποστηρίζει ολοκληρωμένα μοντέλα βαθιάς μάθησης. Υποστηρίζει περίπου 40 δημόσια μοντέλα και περίπου 40 προ-εκπαιδευμένα μοντέλα Intel μέσω της Intel Model Zoo. Το μοντέλο «emotion-recognition-retail-0003» είναι ένα πλήρες δίκτυο συνέλιξης για την αναγνώριση πέντε συναισθημάτων («ουδέτερο», «χαρούμενο», «θλίψη», «έκπληξη», «θυμός»). Για την αξιολόγηση των μετρήσεων, χρησιμοποιείται το τμήμα επικύρωσης του συνόλου δεδομένων AffectNet.

4.1.4 MobileNet

Το MobileNet είναι ένα είδος συνθετικού νευρικού δικτύου σχεδιασμένο από έρευνες της Google. Έχει δημιουργηθεί για “κινητά πρώτα”, επειδή είναι αρχιτεκτονικά σχεδιασμένο για να είναι φιλικό προς τους πόρους χαμηλής απόδοσης και τρέχει γρήγορα, στο κινητές συσκευές.

4.2 Βιβλιοθήκες μηχανικής όρασης - OpenCV

Το OpenCV είναι μια βιβλιοθήκη ανοιχτού κώδικα, μηχανικής όρασης και λογισμικού μηχανικής μάθησης. Δημιουργήθηκε για να παρέχει μια κοινή υποδομή για εφαρμογές μηχανικής όρασης και να επιταχύνει τη χρήση της αντίληψης της σε εμπορικά προϊόντα. Τα κύρια χαρακτηριστικά του είναι:

- Μεγάλο εύρος υποστήριξης σε γλώσσες προγραμματισμού : Υποστηρίζει C ++, Python, Java, Javascript και MATLAB.
- Πληθώρα Αλγορίθμων: Παρέχει πρόσβαση σε περισσότερους από 2.500 αλγορίθμους, έτοιμους για χρήση.
- Είναι δοκιμασμένο: Μεγάλες εταιρείες όπως η IBM, η Google, η Toyota χρησιμοποιούν το OpenCV για πολλαπλά καθήκοντα. Με αυτόν τον τρόπο, οι χρήστες είναι βέβαιοι ότι έχουν πρόσβαση σε μια βιβλιοθήκη που χρησιμοποιείται από κυβερνητικά ιδρύματα και επιχειρήσεις.
- Μεγάλη κοινότητα: Στη μεγάλη κοινότητα του OpenCV, οι χρήστες μπορούν να ζητήσουν βοήθεια και να βοηθήσουν άλλους προγραμματιστές. Αυτό δίνει στους προγραμματιστές πρόσβαση σε πληροφορίες σχετικά με τη βιβλιοθήκη και τους κωδικούς.
- Αποτελεσματική λύση: Το OpenCV παρέχει αλγοριθμική απόδοση κυρίως για την επεξεργασία προγραμμάτων σε πραγματικό χρόνο. Επιπλέον, έχει σχεδιαστεί κατά τρόπο που να επιτρέπει την αξιοποίηση της επιτάχυνσης υλικού και των συστημάτων πολλαπλών πυρήνων για ανάπτυξη. Το OpenCV, εμπεριέχει την υλοποίηση της αναγνώρισης προσώπων χρησιμοποιώντας ταξινομητές κλιμάκωσης βάση χαρακτηριστικών Haar, που είναι μια αποτελεσματική μέθοδος μηχανικής μάθησης με 95% ακρίβεια.

Για την εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη OpenCV.js: Η βιβλιοθήκη OpenCV φορτώνεται μεταγλωττισμένη και ετεροχρονισμένα διότι είναι μεγάλο αρχείο. Χρησιμοποιήθηκε το Localforage 1.7.3.js που είναι είναι μια βιβλιοθήκη Javascript, που χειρίζεται την cache μνήμη του περιηγητή στον δίσκο. Έτσι μπορούν να φορτωθούν μεγάλες βιβλιοθήκες από την cache, χωρίς να χρειαστεί να ξανακατέβουν από το δίκτυο.

Μέσω της βιβλιοθήκης localforage, η βιβλιοθήκη OpenCV.js γίνεται cached και δεν χρειάζεται να ξανακατέβει, στην επόμενη είσοδο στον ιστότοπο.

4.3 Τεχνολογίες ανάπτυξης λογισμικού

Για την ανάπτυξη λογισμικού χρησιμοποιήθηκαν πλαίσια προγραμματισμού που επιτρέπουν την ταυτόχρονη ανάπτυξη εφαρμογών, που λειτουργεί σε προγράμματα περιήγησης και κινητές συσκευές. Διακρίθηκαν δύο από τα πιο σύγχρονα και εύχρηστα μοντέλα: Ionic και React.

4.3.1 Ionic

Θετικά χαρακτηριστικά:

- Είναι ελεύθερο και βασισμένο σε ανοικτό κώδικα
- Είναι εύκολο στην εκμάθηση
- Έχει καθαρή και ενημερωμένη τεκμηρίωση
- Δημιουργεί εφαρμογές για iOS, Android, Windows, Desktop, Web και PWA Αρνητικά χαρακτηριστικά:
- Δεν έχει σταθερά καλή και γρήγορη απόδοση
- Χρειάζεται πρόσθετα για να λειτουργήσει σε πολλές διαφορετικές συσκευές
- Χρησιμοποιεί το WebView για να προσομοιώσει την τοπική λειτουργικότητα εφαρμογής σε κινητές συσκευές.

4.3.2 React

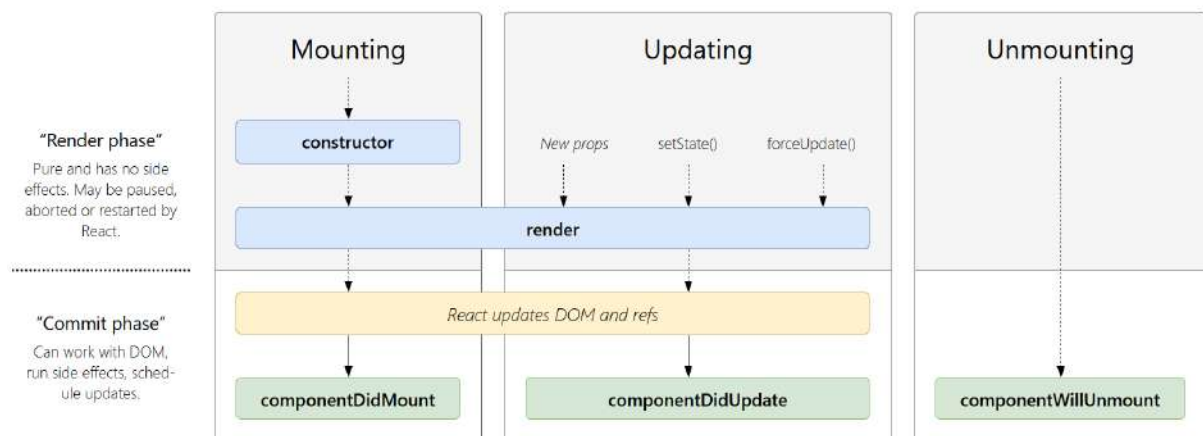
Θετικά χαρακτηριστικά:

- Είναι ελεύθερο και βασισμένο σε ανοικτό κώδικα
- Μέγιστη επαναχρησιμοποίηση κώδικα για την ανάπτυξη εφαρμογών για πλατφόρμες Android, iOS και Windows
- Σταθερή πλατφόρμα για την ανάπτυξη μεγάλων έργων

- Ταχεία ανάπτυξη με αποτελεσματική δομή κώδικα
 - Ευρεία κοινότητα και λειτουργεί ανεξάρτητα από την πλατφόρμα του κώδικα
- Αρνητικά χαρακτηριστικά:
- Χρειάζεται αρκετή εκμάθηση
 - Δεν έχει πρόσθετα

Έτσι επιλέχτηκε η React, διότι ικανοποιεί τις ανάγκες της εφαρμογής για γρήγορη ανάπτυξη και συμβατότητα με πολλές συσκευές.

Η βιβλιοθήκη React, κατασκευάζει στοιχεία (components) που χρησιμοποιούνται ως κλάσεις. Η κάθε τέτοια κλάση πρέπει να υλοποιεί τουλάχιστον μια συνάρτηση, την Απεικόνιση (Render).



Σχήμα 4.1: Τρόπος λειτουργίας React Component

Το κάθε component, έχει 3 καταστάσεις (states). Η πρώτη κατάσταση είναι όταν δημιουργήθηκε για πρώτη φορά στο DOM. Όταν τα στοιχεία του στοιχείου μεταβάλλονται, καλείται η ενημέρωση και τέλος, όταν πρέπει να καταστραφεί, καλείται η αποσύνδεση. Η εφαρμογή, που είναι χτισμένη σε React δημιουργεί τις εξής βασικές κλάσεις:

- **Video Room Component:** Είναι η κλάση υπεύθυνη για την δημιουργία σύνδεσης, της ζωντανής μετάδοσης και λήψης βίντεο με τους συνομιλητές που βρίσκονται στο ίδιο δωμάτιο. Αυτό το στοιχείο επιτρέπει στο χρήστη να σβήνει το μικρόφωνο, να καταργεί τη δημοσίευση της κάμερας, να μοιράζεται την οθόνη, να ανοίγει τη συνομιλία και να αποσυνδέεται από την συνομιλία.

- **Stream Component:** Είναι η κλάση που μαζί με την OpenVidu Layout, είναι υπεύθυνη για την εμφάνιση της ροής βίντεο κάθε χρήστη. Επειδή οι ροές βίντεο είναι δυναμικές, πρέπει να εμφανίζονται και στην οθόνη του χρήστη με έναν δυναμικό τρόπο.
- **Video Component:** Είναι η κλάση υπεύθυνη για το ίδιο το βίντεο. Περιέχει το βίντεο ως αντικείμενο μαζί με τα χαρακτηριστικά του.

4.4 Πρωτόκολλα και Βιβλιοθήκες δικτύου

4.4.1 Πρωτόκολλο MQTT

Το MQTT (Message Queue Telemetry Transport) πρόκειται για ένα εκδοτικό / συνδρομητικό, εξαιρετικά απλό και ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων, σχεδιασμένο για περιορισμένες συσκευές και δίκτυα χαμηλού εύρους ζώνης, υψηλής καθυστέρησης ή αναξιόπιστων δικτύων. Οι αρχές σχεδιασμού είναι να ελαχιστοποιηθούν οι απαιτήσεις για το εύρος ζώνης δικτύου και τους πόρους των συσκευών, ενώ ταυτόχρονα επιχειρείται η διασφάλιση της αξιοπιστίας και η έγκυρη παράδοση. Αυτές οι αρχές καταδεικνύουν ότι το πρωτόκολλο είναι ιδανικό για τον κόσμο των συνδεδεμένων συσκευών σε επικοινωνίες "μηχανής σε μηχανής" (M2M) ή "Internet of Things" και για κινητές εφαρμογές όπου το εύρος ζώνης και η ισχύς της μπαταρίας είναι δεν εξαιρετικά υψηλές. Το MQTT είναι ένα ελαφρύ και ευέλικτο πρωτόκολλο δικτύου που επιτυγχάνει το σωστό ισοζύγιο για συσκευές IOT:

- Το ελαφρύ πρωτόκολλο επιτρέπει την υλοποίησή του τόσο σε σταθερό δίκτυο συσκευών όσο και σε δίκτυα υψηλού λανθάνοντος / περιορισμένου εύρους ζώνης.
- Η ευελιξία του επιτρέπει την υποστήριξη ποικίλων σεναρίων εφαρμογών για συσκευές και υπηρεσίες IoT.

4.4.2 Σύγκριση MQTT με HTTP

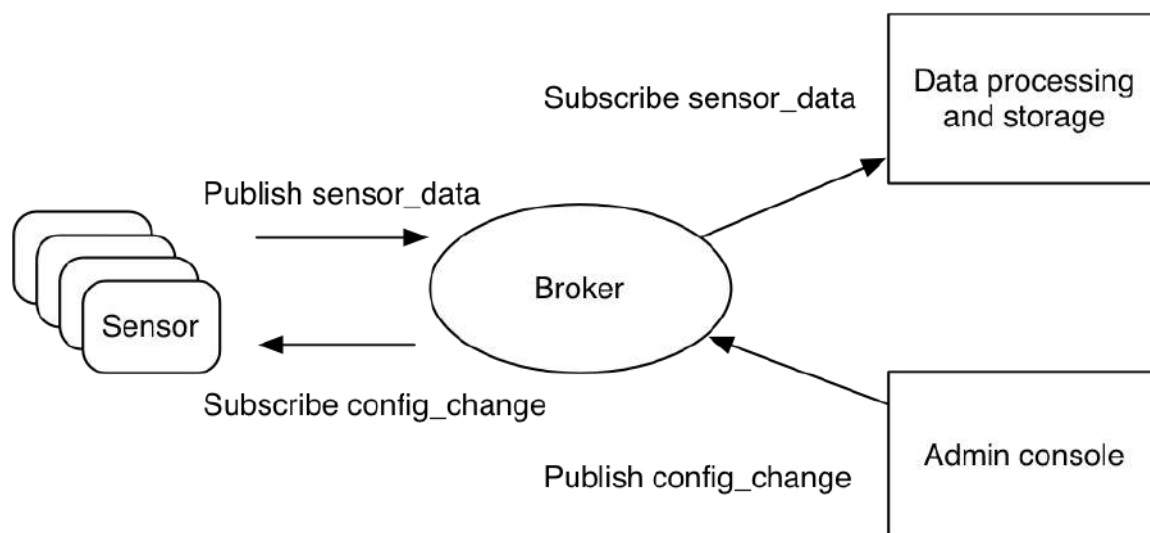
Το HTTP είναι ένα σύγχρονο πρωτόκολλο. Ο υπολογιστής-πελάτης περιμένει να ανταποκριθεί ο διακομιστής. Αυτή είναι μια απαίτηση για προγράμματα περιήγησης ιστού, αλλά έρχεται με το κόστος της κακής κλιμάκωσης. Στον κόσμο των IoT, ο μεγάλος αριθμός συσκευών και, πιθανότατα, ένα αναξιόπιστο / υψηλό λανθάνον δίκτυο

έχουν καταστήσει την σύγχρονη επικοινωνία προβληματική. Ένα ασύγχρονο πρωτόκολλο ανταλλαγής μηνυμάτων είναι πολύ πιο κατάλληλο για εφαρμογές IoT. Οι αισθητήρες μπορούν να στείλουν τις αναγνώσεις και να επιτρέψουν στο δίκτυο να υπολογίσει τη βέλτιστη διαδρομή και το χρονοδιάγραμμα για την παράδοση στις συσκευές και τις υπηρεσίες προορισμού του.

- Το HTTP είναι μονόδρομο. Ο πελάτης πρέπει να ξεκινήσει τη σύνδεση. Σε μια εφαρμογή IoT, οι συσκευές ή οι αισθητήρες είναι συνήθως πελάτες, πράγμα που σημαίνει ότι δεν μπορούν να λαμβάνουν παθητικά εντολές από το δίκτυο.
- Το HTTP είναι ένα πρωτόκολλο 1-1. Ο πελάτης υποβάλλει ένα αίτημα και ο διακομιστής αποκρίνεται. Είναι δύσκολο και ακριβό να μεταδίδεται ένα μήνυμα σε όλες τις συσκευές του δικτύου, κάτι που αποτελεί κοινή περίπτωση χρήσης στις εφαρμογές IoT.
- Το HTTP είναι ένα βαρύ πρωτόκολλο με πολλές κεφαλίδες και κανόνες. Δεν είναι κατάλληλο για περιορισμένα δίκτυα.
- Το MQTT είναι ελαφρύ. Έχει μια απλή κεφαλίδα για να καθορίσει τον τύπο του μηνύματος, ένα θέμα βασισμένο σε κείμενο και στη συνέχεια ένα αυθαίρετο δυαδικό ωφέλιμο φορτίο. Η εφαρμογή μπορεί να χρησιμοποιήσει οποιαδήποτε μορφή δεδομένων για το ωφέλιμο φορτίο, όπως JSON, XML, κρυπτογραφημένο δυαδικό ή Base64, εφόσον οι πελάτες προορισμού μπορούν να αναλύσουν το ωφέλιμο φορτίο.

4.4.3 Τρόπος Λειτουργίας MQTT

1. Με το MQTT ο πελάτης συνδέεται με τον μεσίτη. Μπορεί να εγγραφεί σε οποιοδήποτε μήνυμα "θέμα" στο μεσίτη. Αυτή η σύνδεση μπορεί να είναι απλή σύνδεση TCP / IP ή κρυπτογραφημένη σύνδεση TLS για ευαίσθητα μηνύματα.
2. Ο πελάτης δημοσιεύει μηνύματα κάτω από ένα θέμα στέλνοντας το μήνυμα και το θέμα στον μεσίτη.
3. Ο μεσίτης προωθεί το μήνυμα σε όλους τους πελάτες που προσυπογράφουν το συγκεκριμένο θέμα.



Σχήμα 4.2: Το μοντέλο MQTT δημοσιεύει και εγγράφει μηνύματα από και προς αισθητήρες IoT

4.4.4 Mosquitto Broker

Το Eclipse Mosquitto είναι ένας μεσίτης μηνυμάτων ανοιχτού κώδικα που υλοποιεί τις εκδόσεις πρωτοκόλλων MQTT 5.0, 3.1.1 και 3.1. Το Mosquitto είναι ελαφρύ και είναι κατάλληλος για χρήση σε όλες τις συσκευές από υπολογιστές με χαμηλή ισχύ σε single-board σε διακομιστές.

4.4.5 Paho Βιβλιοθήκη

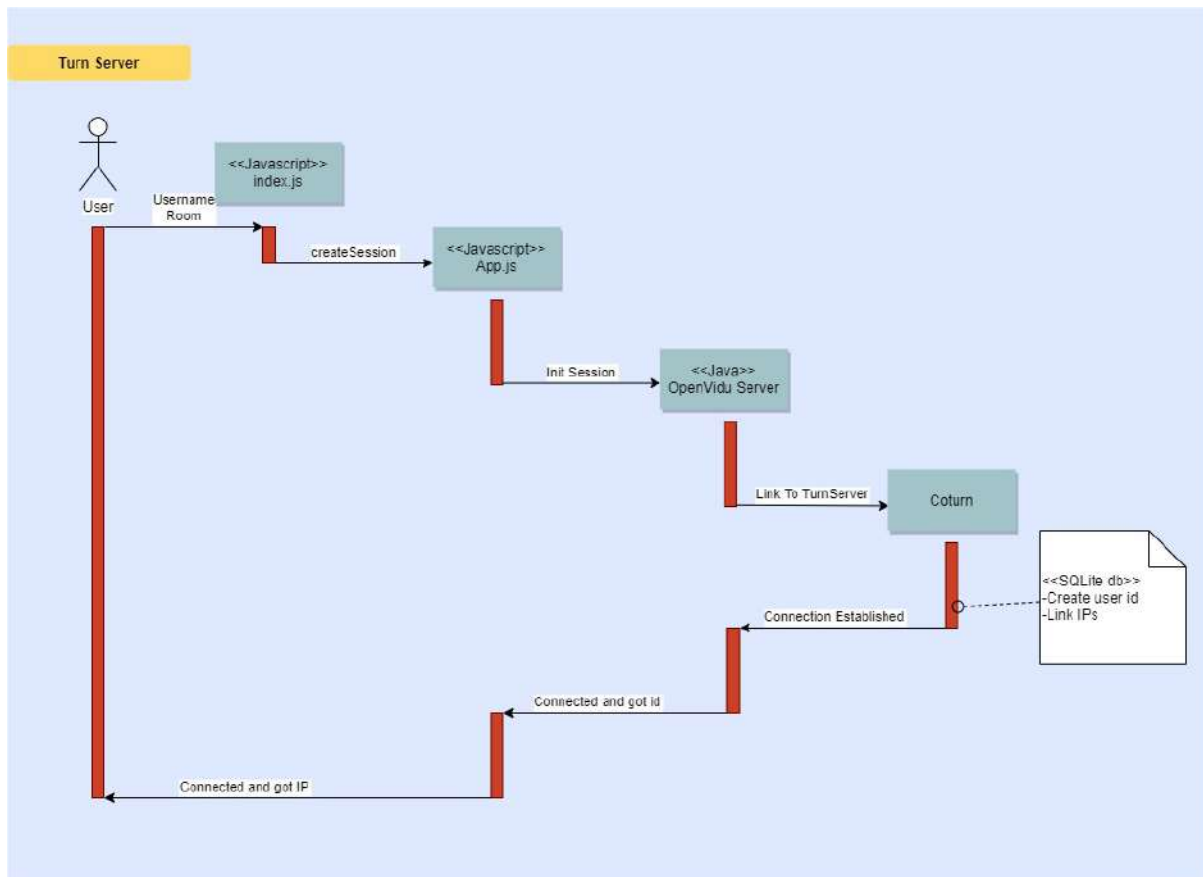
Το πρόγραμμα Eclipse Paho παρέχει αξιόπιστες εφαρμογές ανοιχτού κώδικα ανοικτών και τυπικών πρωτοκόλλων ανταλλαγής μηνυμάτων που στοχεύουν σε νέες, υπάρχουσες και αναδυόμενες εφαρμογές για Machine to Machine (M2M) και Internet of Things. Το Paho είναι φτιαγμένο τους εγγενείς φυσικούς και οικονομικούς περιορισμούς της συνδεσιμότητας των συσκευών. Περιέχει υλοποιήσεις πελατών MQTT publish / subscribe για χρήση σε ενσωματωμένες πλατφόρμες, μαζί με την αντίστοιχη υποστήριξη διακομιστή όπως καθορίζεται από την κοινότητα. Επιπλέον, χρησιμοποιεί TLS αντί για απλό TCP. Στην εφαρμογή χρησιμοποιήθηκε το mqttws31.min.js, υλοποίηση του Paho στην γλώσσα προγραμματισμού Javascript.

5 Λειτουργικές Απαιτήσεις και περιπτώσεις χρήσης

5.1 Λειτουργικές απαιτήσεις συστήματος

1. Ο χρήστης πρέπει να εισαχθεί στο σύστημα μέσω ενός περιηγητή ιστού.
2. Ο χρήστης πρέπει να εισάγει αναγνωριστικό χρήστη και το εικονικό δωμάτιο, για να αντιστοιχηθεί
3. Δυναμική δημιουργία χρήστη και εικονικού δωματίου.
4. Δημιουργία εκπομπής και λήψης βίντεο προς και από τους άλλους χρήστες στο ίδιο δωμάτιο.
5. Ο χρήστης επιλέγει μοντέλο μηχανικής μάθησης, μέγιστο αριθμό εικόνων προς ανάλυση το δευτερόλεπτο και την λειτουργία εγγραφής.
6. Εκκίνηση μηχανικής μάθησης σε ένα επιλεγμένο βίντεο.
7. Εξαγωγή αποτελέσματος ανάλυσης και εμφάνιση του στον χρήστη.
8. Διακοπή της διαδικασίας ανάλυσης της εικόνας και εκτύπωση των χρόνων.
9. Αλλαγή του βίντεο προς ανάλυση.
10. Αλλαγή του τρόπου ανάλυσης του βίντεο.
11. Έξοδος του χρήστη από το σύστημα.
12. Αποκοπή δεδομένων και επεξεργασία τους για την εισαγωγή τους στους αλγόριθμους μηχανικής μάθησης.
13. Εξωτερικός διακομιστής για την απομακρυσμένη ανάλυση των βίντεο.
14. Αποδοχή από τον χρήστη των πιστοποιητικών ιστού.
15. Δημιουργία δικτυακής τοπολογίας για την επικοινωνία των χρηστών.

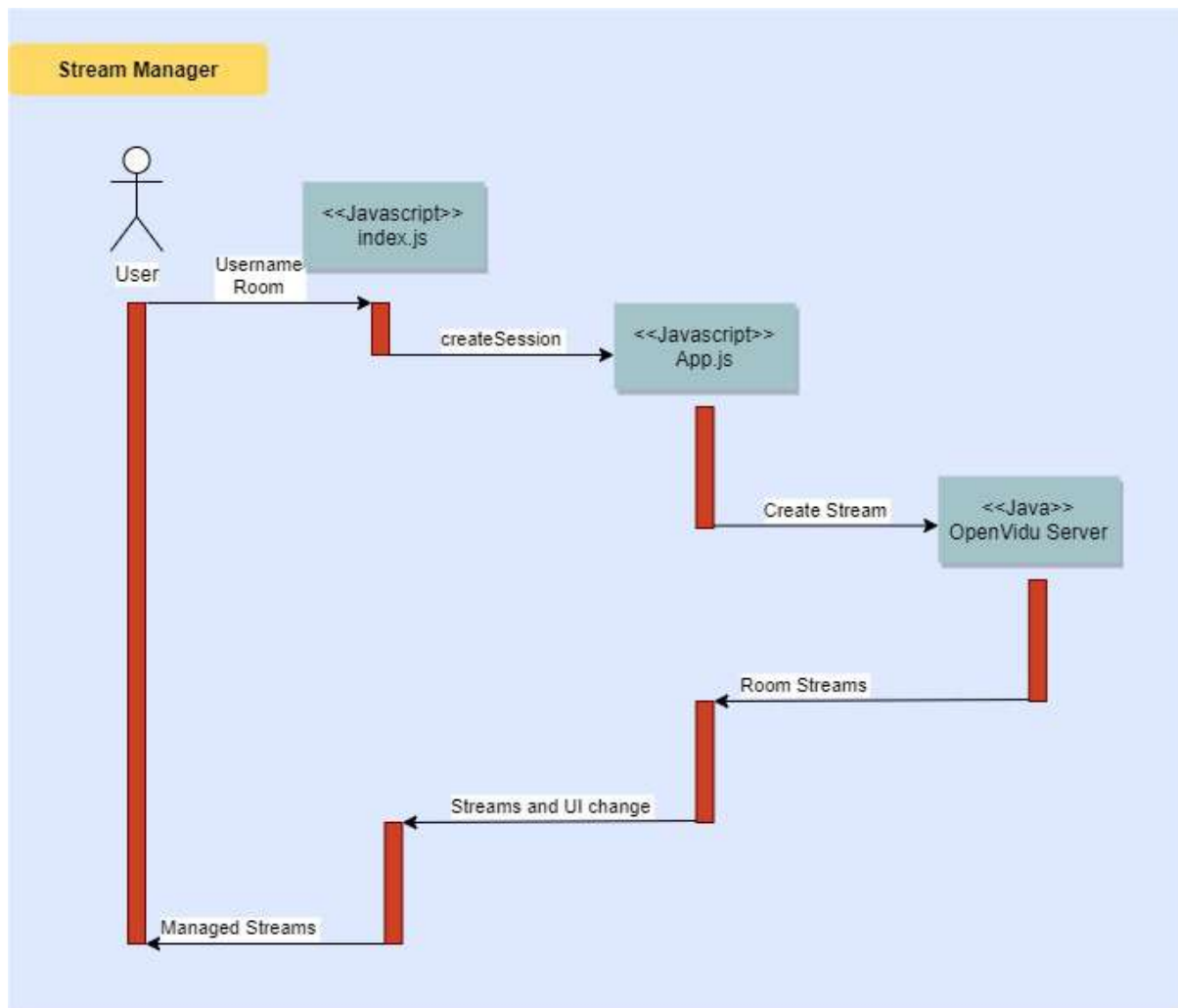
5.2 Διαγράμματα ροής



Σχήμα 5.1: Διευθυνσιοδότηση με τον διακομιστή ICE/TURN

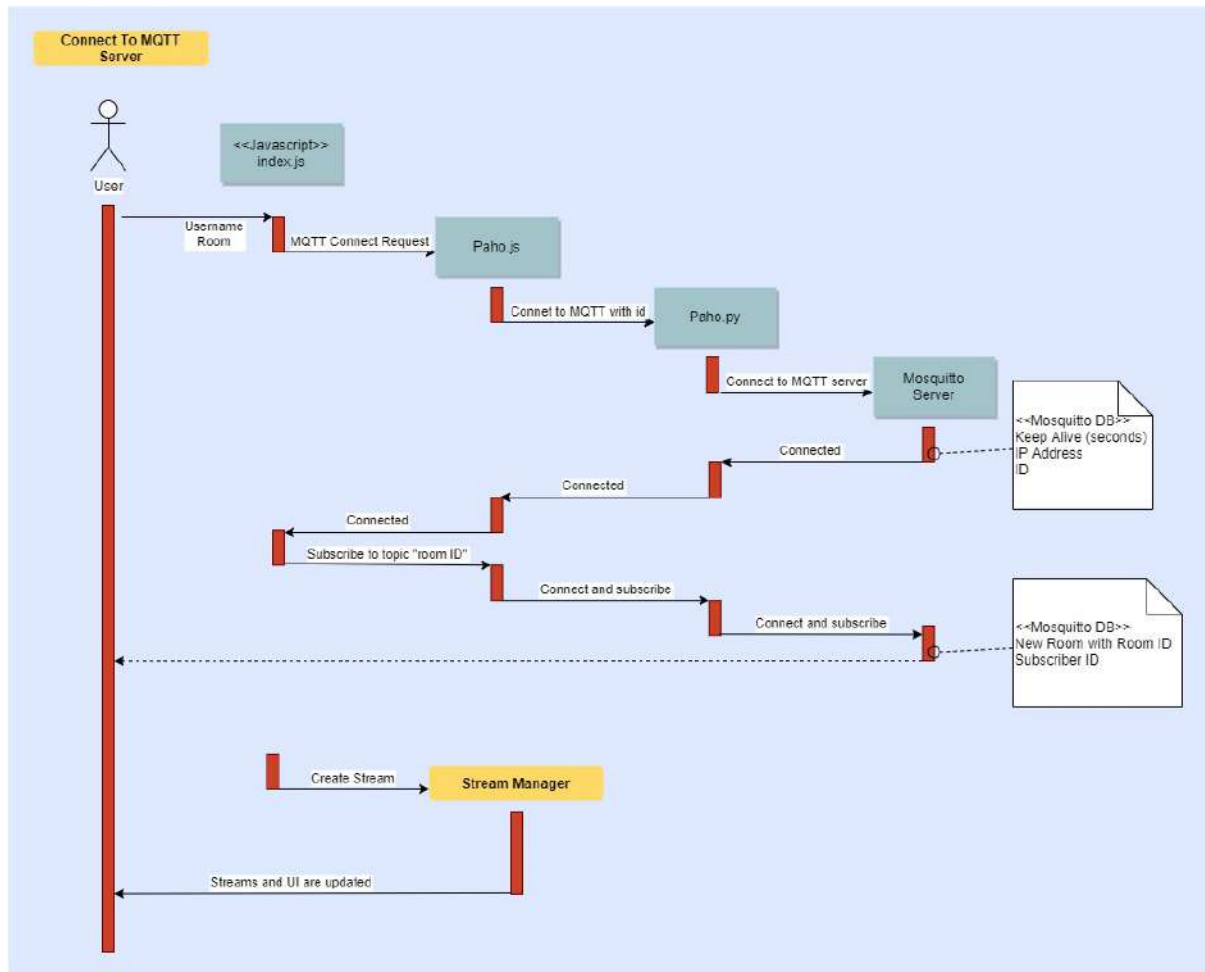
Σε αυτό το διάγραμμα απεικονίζεται η πρώτη πράξη που γίνεται όταν ο χρήστης εισέρχεται στην πλατφόρμα.

1. Αρχικά διαλέγει όνομα χρήστη και δωμάτιο
2. Έπειτα εκκινείται στο backend η διαδικασία εκκίνησης συνεδρίας.
3. Αποστέλλονται τα δεδομένα στον εξυπηρετητή ICE/TURN και εκείνος επιστρέφει μια IP, που θα χρησιμοποιηθεί από εδώ και πέρα στην εφαρμογή για την επικοινωνία με τον χρήστη.



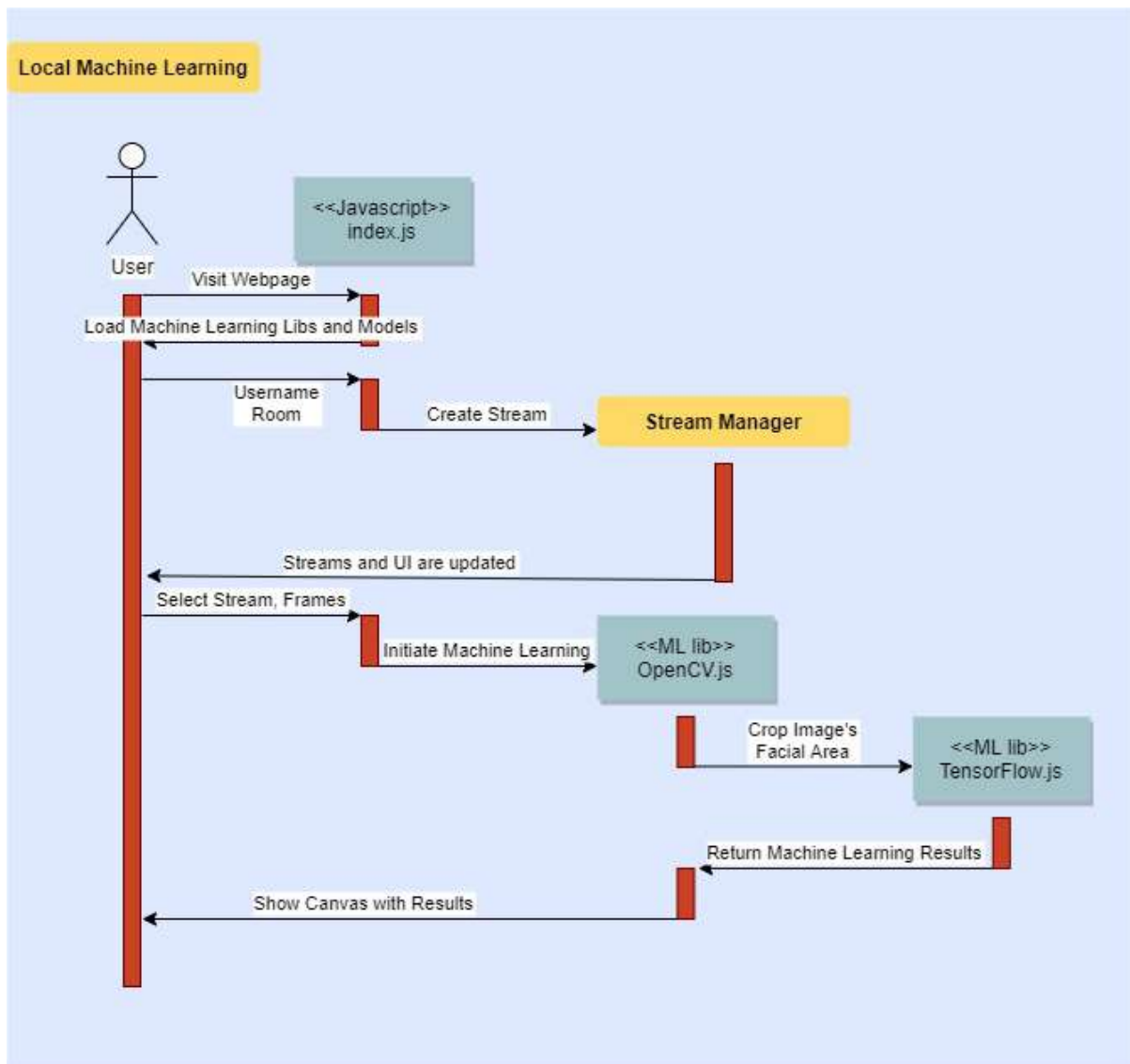
Σχήμα 5.2: Διαχείριση βίντεο-μεταδόσεων

1. Ο χρήστης εισέρχεται στο σύστημα, και βάζει στοιχεία χρήστη και το δωμάτιο που επιλέγει.
2. Δημιουργείται μια συνεδρία με βάση το δωμάτιο.
3. Εκκινείται η διαδικασία της μετάδοσης βίντεο.
4. Ο διακομιστής OpenVidu επιστρέφει όλα τα βίντεο για αυτό το δωμάτιο στον χρήστη.



Σχήμα 5.3: Σύνδεση στο σύστημα MQTT

1. Ο χρήστης εισέρχεται στο σύστημα, και βάζει στοιχεία χρήστη και το δωμάτιο που επιλέγει.
2. Έπειτα Επικοινωνεί με το σύστημα Paho MQTT.
3. Του δίνεται ένα τυχαίο ψευδώνυμο και γίνεται η μεταφορά της πληροφορίας (ψευδώνυμο, Διεύθυνση IP) προς τον εξυπηρετητή Paho.
4. Ο εξυπηρετητής Paho, αναλαμβάνει την μετάδοση της πληροφορίας στον διακομιστή Mosquitto.
5. Ο εξυπηρετητής Mosquitto, αποθηκεύει τον χρήστη με επικεφαλίδα το δωμάτιο και του δίνει δικαιώματα να λάβει και να στείλει μηνύματα.

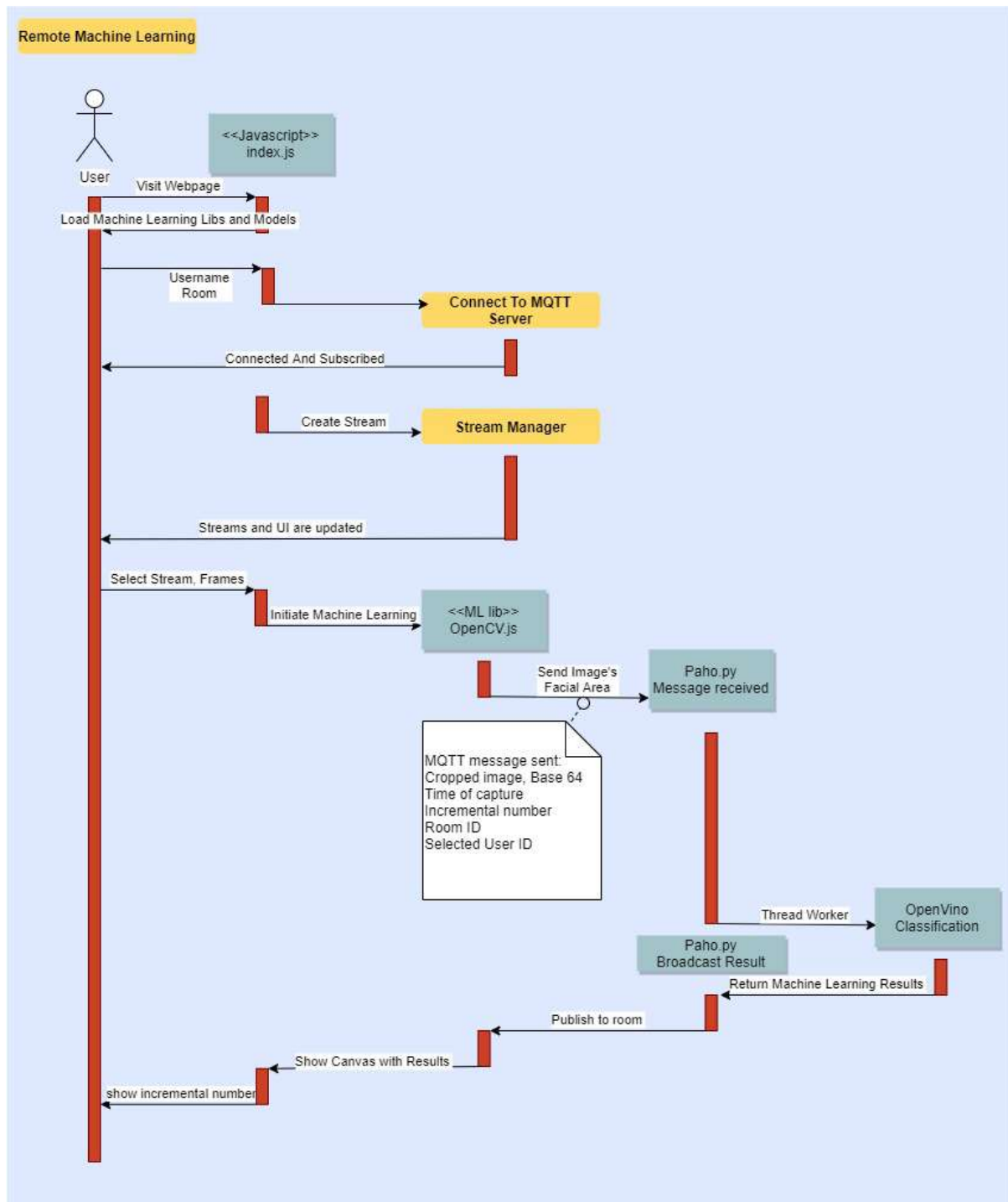


Σχήμα 5.4: Τοπική μηχανική μάθηση

1. Ο χρήστης εισέρχεται στην εφαρμογή. Απευθείας φορτώνονται οι βιβλιοθήκες μηχανικής μάθησης και τα μοντέλα, για την επερχόμενη χρήση τους.
2. Ο χρήστης βάζει στοιχεία χρήστη και το δωμάτιο που επιλέγει.
3. Δημιουργούνται οι διαθέσιμες βίντεο μεταδόσεις.
4. Επιλέγει ο χρήστης ως παραμέτρους σε ποιες βίντεο μεταδόσεις και πόσες αναλύσεις το δευτερόλεπτο θα πραγματοποιήσει ο αλγόριθμος.
5. Έστερα μέσω της βιβλιοθήκης OpenCV θα γίνει εγγραφή μιας εικόνας του βίντεο και θα αποκοπεί το μέρος του προσώπου.
6. Θα δώσει το αποτέλεσμα στο μοντέλο μηχανικής μάθησης, βασισμένο στο TensorFlow

για πρόβλεψη συναισθήματος.

7. Τέλος θα επιστρέψει το αποτέλεσμα πίσω στον χρήστη.



Σχήμα 5.5: Απομακρυσμένη μηχανική μάθηση

1. Ο χρήστης εισέρχεται στην εφαρμογή. Απευθείας φορτώνονται οι βιβλιοθήκες μηχανικής μάθησης και τα μοντέλα, για την επερχόμενη χρήση τους.

2. Ο χρήστης βάζει στοιχεία χρήστη και το δωμάτιο που επιλέγει.
3. Δημιουργούνται οι συνδέσεις με τον χειριστή μηνυμάτων MQTT.
4. Δημιουργούνται οι διαθέσιμες βίντεο μεταδόσεις.
5. Επιλέγει ο χρήστης ως παραμέτρους σε ποιες βίντεο μεταδόσεις και πόσες αναλύσεις το δευτερόλεπτο θα πραγματοποιήσει ο αλγόριθμος.
6. Έστερα μέσω της βιβλιοθήκης OpenCV θα γίνει εγγραφή μιας εικόνας του βίντεο και θα αποκοπεί το μέρος του προσώπου.
7. Συγγράφει ένα αρχείο με τα δεδομένα της εικόνας (κωδικοποιημένη σε συμβολοσειρά), του χρόνου, του αριθμού επανάληψης, του δωματίου, και του ID χρήστη που γίνεται η ανάλυση.
8. Αυτό μεταφέρεται σαν μήνυμα μέσω του Paho, και προωθείται στον μεσίτη MQTT.
9. Ο μεσίτης MQTT, μεταδίδει τα αποτελέσματα πίσω, σε όλου τους χρήστες του δωματίου.

6.1 Επισκόπηση της διαδικασίας

6.1.1 Ubuntu εξυπηρετητής

Στον server έχουν εγκατασταθεί ο KMS, το Coturn, ο διακομιστής OpenVidu και ο εξυπηρετητής της εφαρμογής. Ρόλος του server είναι:

- Να συνδέονται οι πελάτες στον webserver, για να εκκινήσουν την διαδικασία της μετάδοσης και παρακολούθησης βίντεο.
- Ο διακομιστής Web, να διαχειρίζεται τους χρήστες και τα “δωμάτια” στα οποία ο κάθε χρήστης εισέρχεται.
- Να συνδέεται ο κάθε πελάτης με τον OpenVidu server και τον KMS, για την διαχείριση του όσον αφορά το πρωτόκολλο του WebRTC και τις μεταδώσεις των βίντεο.
- Εάν ο πελάτης βρίσκεται πίσω από NAT, τότε ο KMS αναλαμβάνει να δημιουργήσει μια εξωτερική, ανεξάρτητη σύνδεση, μέσω του πρωτοκόλλου ICE/TURN.
- Το Coturn διατηρεί μια Sqlite βάση δεδομένων με τους ζωντανούς χρήστες και την IP τους.
- OpenVidu-server: εφαρμογή Java που ενορχηστρώνει τον Kurento Media Server.

6.1.2 Εφαρμογή Πελάτη

- Η εφαρμογή του χρήστη σερβίρεται από τον WebServer. Ο χρήστης πρέπει να έχει πρόσβαση στην IP/θύρα του webserver, για να εισαχθεί στην εφαρμογή.
- Ο πελάτης συνδέεται με το OpenVidu server και είναι έτοιμος για την έναρξη της ενδοεπικοινωνίας.
- Αν πάσα στιγμή, ο πελάτης μπορεί να ενεργοποιήσει την μηχανική μάθηση σε ένα βίντεο.
- Εφόσον ξεκινήσει η μηχανική μάθηση σε ένα βίντεο το Paho client, αναλαμβάνει να αποστείλει και να λάβει την ανάλυση των frames, από το OpenVino.
- OpenVidu-browser: Βιβλιοθήκη Javascript για το πρόγραμμα περιήγησης. Επιτρέπει την άμεση διαχείριση των βιντεοκλήσεων από τους χρήστες.

6.1.3 Ubuntu εξυπηρετητής με Intel επεξεργαστή

- Υπεύθυνος για την λειτουργία του MQTT είναι το Mosquitto.
- Το Paho, διαχειρίζεται τα μηνύματα που έρχονται από το Paho client και τα προωθεί στο OpenVino. Μετά λαμβάνει την απάντηση και την στέλνει πίσω στους Paho clients.
- Το OpenVino είναι υπεύθυνο για την μηχανική μάθηση στον server.

6.2 Εκκίνηση/Ακολουθία μια βιντεοδιάλεξης

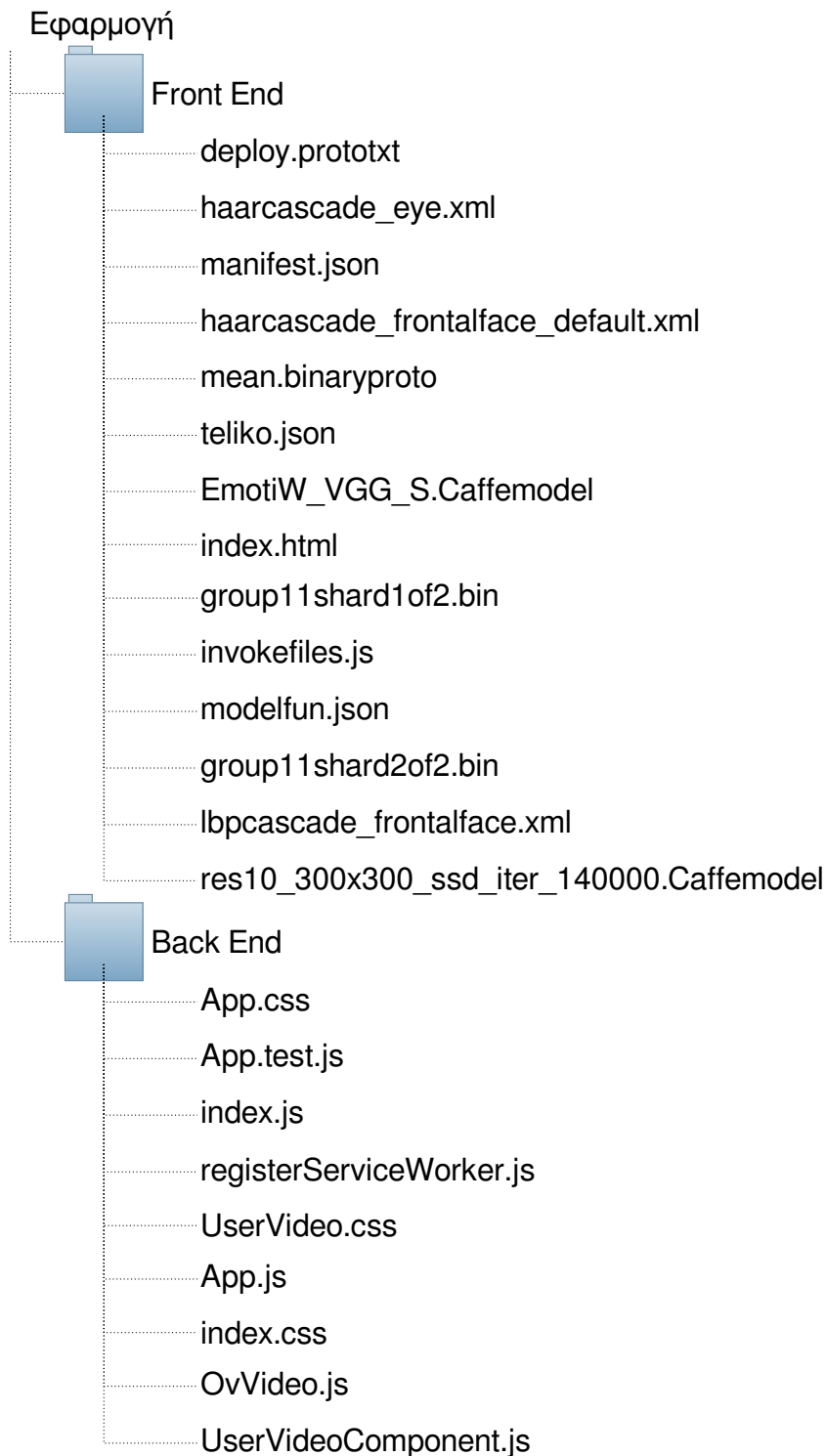
6.3 Δομή Αρχείων

Τα βασικά αρχεία της εφαρμογής είναι:

- OpenVidu-browser.js: (OpenVidu).
- app.js: Επεκτάθηκε έτσι ώστε να υπάρχουν κατάλληλοι μηχανισμοί ελέγχου, ενεργοποίησης και απενεργοποίησης της μηχανικής μάθησης καθώς και η εγγραφή των βίντεο.
- index.html: Επεκτάθηκε έτσι ώστε να ενθυλακώσει την μηχανική μάθηση μέσα στο πρόγραμμα. Είναι το κεντρικό αρχείο για την μηχανική μάθηση καθώς μετά από όλους τους ελέγχους, αναλαμβάνει την λειτουργία του κύριου βρόγχου, ανάλυσης της εικόνας σε κάθε βίντεο. Επιπλέον έγιναν αλλαγές στο πώς εμφανίζονται τα βίντεο στην οθόνη του χρήστη καθώς και αλλαγές που έχουν σκοπό την ομαλή λειτουργία των αλγορίθμων μηχανικής μάθησης με το υπόλοιπο σύστημα.
- style.css: (OpenVidu).
- lbpcascade_frontalface.xml: Αρχείο μηχανικής μάθησης για την αναγνώριση του προσώπου (OpenCV).
- res10_300x300_ssd_iter_140000.Caffemodel: Μοντέλο DNN για την αναγνώριση προσώπου. (OpenCV)
- deploy.prototxt: Αρχείο με την δομή των δεδομένων για την εισαγωγή τους στο μοντέλο του DNN. (OpenCV)

- group11-shard1of2.bin, group11-shard2of2.bin: Αρχεία για φορτώσει το mobilenet. (TensorFlow)
- EmotiW_VGG_S.Caffemodel: Μοντέλο DNN για την αναγνώριση συναισθημάτων. (<https://gist.github.com/GilLevi/54aee1b8b0397721aa4b>)
- mean.binaryproto: Αρχείο μοντέλου Caffee, για την αναγνώριση συναισθημάτων. (<https://gist.github.com/GilLevi/54aee1b8b0397721aa4b>)
- haarcascade_frontalface_default.xml: Μοντέλο για την αναγνώριση προσώπου. (OpenCV)
- haarcascade_eye.xml: Μοντέλο για την αναγνώριση ματιών. (OpenCV)
- modelfun.json: Είναι το αρχείο με τα βάρη για το μοντέλο μηχανικής μάθησης που είναι εκπαιδευμένο με Keras και έχει γίνει μετατροπή για να χρησιμοποιηθεί με TensorFlow. (Μετατροπή από μοντέλο Keras σε TensorFlow)
- teliko.json: Εκπαιδευμένο μοντέλο με transfer learning, πάνω σε συναισθήματα, με MobileNet και TensorFlow.
- OpenCV.js: Βιβλιοθήκη OpenCV σε Javascript. (OpenCV)
- registerServiceWorker.js: Αρχείο OpenVidu για την δημιουργία Web Worker. Χρήσιμο για να σερβίρει παράλληλα με το κύριο νήμα αποθηκευμένα assets της εφαρμογής. (OpenVidu)
- OnVideo.js: Επεκτάθηκε ορίζοντας και την συνάρτηση για το ποια είναι η συμπεριφορά του βίντεο όταν ο χρήστης πατάει πάνω του. Όταν ο χρήστης πατήσει πάνω σε ένα βίντεο τότε σε αυτό γίνεται η εκκίνηση της διαδικασίας μηχανικής μάθησης. Επιπλέον εάν αυτό δεν είναι ήδη το κεντρικό βίντεο τότε μεταφέρεται στο κεντρικό. Το αντικείμενο βίντεο έχει ως χαρακτηριστικά: Το μοναδικό αναγνωριστικό του. Τις διαστάσεις του στην ιστοσελίδα. Το τοπικό αναγνωριστικό του. Συνοδεύεται από μια συνάρτηση που ενεργοποιεί και απενεργοποιεί την εκκίνηση της μηχανικής μάθησης πάνω στο βίντεο.
- UserVideoComponent.js: Επεκτάθηκε έτσι ώστε να δίνει το αναγνωριστικό του χρήστη του βίντεο, όταν υπάρξει αλλαγή στο κεντρικό βίντεο.

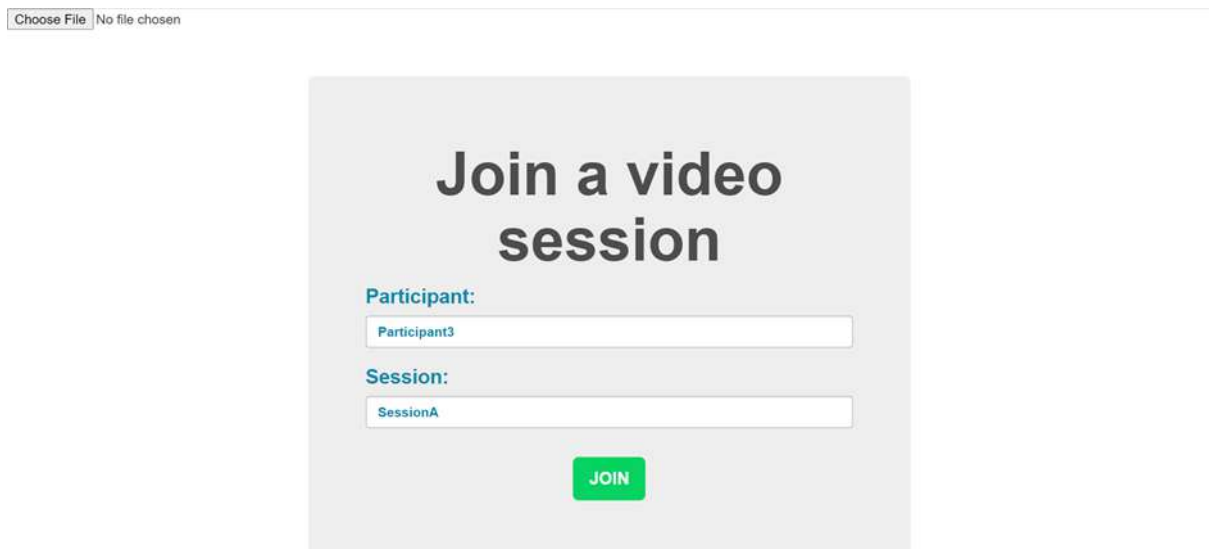
- UserVideo.css: Αρχείο που περιέχει λεπτομέρειες για το πώς θα εμφανιστεί το βίντεο του χρήστη στον χρήστη. (OpenVidu)
- index.js: Αρχείο εκκίνησης της εφαρμογής. (OpenVidu)
- index.css: Αρχείο που περιέχει κανόνες για την εμφάνιση της εφαρμογής σε υψηλό επίπεδο. (OpenVidu)



7 Λειτουργικότητα

7.1 Αρχική Σελίδα

Η αρχική σελίδα είναι η πρώτη σελίδα της πλατφόρμας.

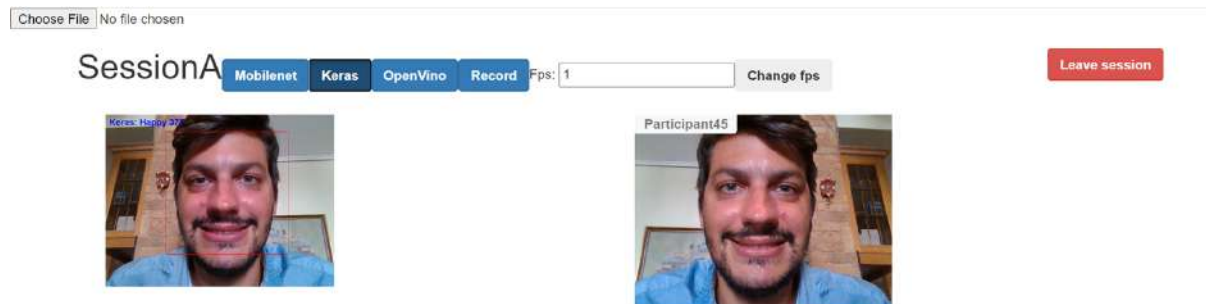


Σχήμα 7.1: Αρχική σελίδα

Ο χρήστης στην αρχική σελίδα εισάγει το όνομα του κάτω από το πεδίο "participant". Ο ρόλος του ονόματος είναι για την μοναδικότητα κάθε χρήστη μέσα στο σύστημα, καθώς και για την εύρεση του βίντεο του στην μηχανική μάθηση. Προκαθορισμένα το όνομα του χρήστη είναι "participant" ακολουθούμενο από ένα τυχαίο νούμερο από 0 μέχρι 99.

Έπειτα ο χρήστης εισάγει το δωμάτιο "Session", που θέλει να συμμετάσχει. Ο χρήστης μπορεί να εισάγει οποιοδήποτε όνομα δωματίου επιθυμεί και θα εισαχθεί σε αυτό το δωμάτιο. Για να γίνει μετάδοση βίντεο από τον ένα χρήστη σε έναν άλλο πρέπει και οι δύο να έχουν επιλέξει το ίδιο δωμάτιο. Προκαθορισμένα το όνομα του δωματίου είναι "SessionA". Επιπλέον υπάρχει ένα κουμπί με την ονομασία "browse" για να εισαχθεί ένα μοντέλο μηχανικής μάθησης από τον δίσκο.

7.2 Κεντρική Σελίδα

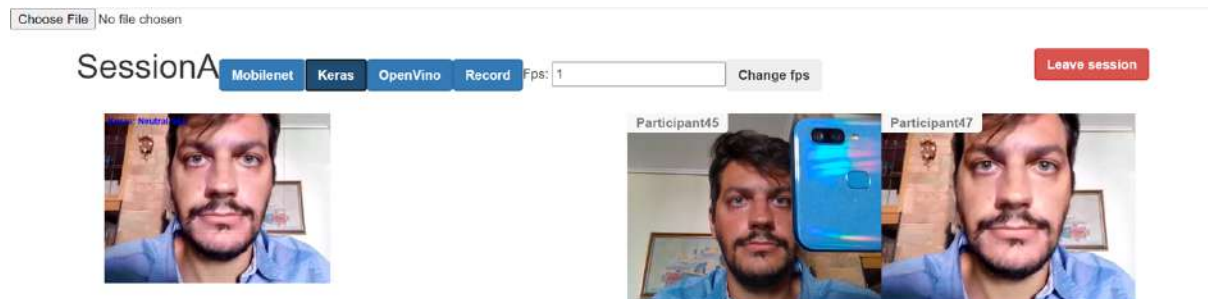


Σχήμα 7.2: Βίντεο μετάδοση μεταξύ δύο χρηστών

Στην κεντρική σελίδα επέρχονται σχεδόν όλες οι λειτουργίες του προγράμματος που είναι:

1. Το κουμπί "Browse", όπως και στην αρχική σελίδα.
2. Κουμπί "MobileNet", τύπου ανοιχτό-κλειστό για μηχανική μάθηση με το μοντέλο MobileNet.
3. Κουμπί "OpenVino", τύπου ανοιχτό-κλειστό για μηχανική μάθηση με το μοντέλο OpenVino.
4. Κουμπί "Keras", τύπου ανοιχτό-κλειστό για μηχανική μάθηση με το μοντέλο Keras.
5. Κουμπί "Record", τύπου ανοιχτό-κλειστό για μαγνητοσκόπηση του βίντεο.
6. Πεδίο με αριθμό εικόνων το δευτερόλεπτο. Προεπιλεγμένη τιμή είναι 1 εικόνα το δευτερόλεπτο και κουμπί καταχώρησης "change fps".
7. Κουμπί "Leave Session", εξόδου από την συνεδρία
8. Στα δεξιά είναι τα βίντεο που εκπέμπονται στο δωμάτιο. Πάνω σε κάθε βίντεο είναι το όνομα του χρήστη που συμμετέχει. Το κάθε βίντεο έχει και ένα εσωτερικό αριθμό, που είναι ένας αύξων αριθμός, που μετράει τα βίντεο τα προσδιορίζει με βάση την σειρά που εμφανίζονται. Τα βίντεο είναι εσωτερικά και κουμπιά, τύπου ανοιχτό-κλειστό που όταν πατηθούν ενεργοποιούν την μηχανική μάθηση για αυτό το βίντεο.

9. Στα αριστερά είναι τα βίντεο όπου αναγράφονται τα αποτελέσματα της ανάλυσης της μηχανικής μάθησης. Πάνω αριστερά στο βίντεο ζωγραφίζεται το μοντέλο που έγινε η πρόβλεψη, το αποτέλεσμα της πρόβλεψης και πόσο χρόνος πέρασε για την κάθε πρόβλεψη.



Σχήμα 7.3: Τοπική μηχανική μάθηση

Στο σχήμα φαίνεται η λειτουργία με δύο χρήστες. Η μηχανική μάθηση τρέχει στον χρήστη 45 και τρέχει αλγόριθμο μηχανικής μάθησης με μοντέλο το Keras, για 1 εικόνα το δευτερόλεπτο πάνω στον χρήστη 47.

7.3 Περιγραφή αλγορίθμου Κεντρικής Σελίδας

7.3.1 Πρώτο βήμα - Βιβλιοθήκες και μοντέλα μηχανικής μάθησης

Αρχικά φορτώνονται όλες οι βιβλιοθήκες και τα μοντέλα μηχανικής μάθησης. Επειδή αυτές/αυτά είναι πολλά και μεγάλα αρχεία, φορτώνονται ετεροχρονισμένα. Όταν φορτωθούν εκτυπώνονται μηνύματα στην κονσόλα. Για κάποια αρχεία όπως το OpenCV.js, επειδή είναι πολύ μεγάλο χρησιμοποιείται η βιβλιοθήκη LocalForage. Κάποια αρχεία όπως το μοντέλο μηχανικής μάθησης βασισμένο σε MobileNet, δεν μπορούν να φορτωθούν με LocalForage, λόγω του τρόπου με τον οποίο εισάγονται στο TensorFlow. Έτσι, επειδή το αρχείο είναι 200MB και αργεί να φορτώσει κάθε φορά, χρησιμοποιήθηκε ο τρόπος με την μεταφόρτωση αρχείου.

7.3.2 Δεύτερο βήμα - Αρχικοποιήσεις

Υπάρχουν πολλές μεταβλητές που χρησιμοποιούνται από τα μοντέλα μηχανικής μάθησης, τις αρχικές καταστάσεις των κουμπιών, για την αποθήκευση του τοπικού χρήστη, του τοπικού βίντεο, τα βίντεο που έχει επιλέξει για παρακολούθηση, τον αριθμό επανάληψης κ.α.

7.3.3 Τρίτο βήμα - Σύνδεση με το σύστημα Paho και MQTT

Έπειτα από τις αρχικοποιήσεις το πρόγραμμα θα προσπαθήσει να συνδεθεί μέσω του Paho.js στο Paho.py του διακομιστή. Τα στοιχεία που στέλνει στο διακομιστή είναι ένα ψευδοτυχαίο όνομα και εάν η σύνδεση είναι επιτυχής θα συνδεθεί και θα εγγραφεί στο MQTT με τίτλο το δωμάτιο του. Εάν όχι εκτυπώνεται μήνυμα στην κονσόλα.

7.3.4 Τέταρτο βήμα - Είναι η εκκίνηση της διαδικασίας μηχανικής μάθησης

Σε αυτό το βήματα πρέπει ο χρήστης να έχει πατήσει τουλάχιστον πάνω σε ένα βίντεο για να ξεκινήσει η διαδικασία αναγνώρισης προσώπου από την εικόνα. Με το OpenCV, δημιουργείται ένας πίνακας στις διαστάσεις του βίντεο. Αυτός ο πίνακας θα χρησιμοποιηθεί για να γίνει εγγραφή μιας εικόνας του βίντεο που επέλεξε ο χρήστης. Επιπλέον, επειδή το OpenCV δεν μπορεί να κάνει εγγραφή μιας εικόνας και μετά ο παραγόμενος πίνακας να χρησιμοποιηθεί για να εμφανιστεί το μήνυμα του αποτελέσματος της μηχανικής μάθησης, αντιγράφεται ένα νέο στοιχείο τύπου "canvas", που δεν φαίνεται στον χρήστη, γίνεται η ένωση της εγγεγραμμένης εικόνας με το κείμενο και μεταφέρεται στο τελικό "canvas", που είναι και αυτό που φαίνεται στην εικόνα. Έστερα, θα μπει στο κεντρικό βρόγχο της εφαρμογής.

Algorithm 1: Machine learning in real time video capture

```

Result: Face detection and emotion recognition
Tensorflow initialization;
Opencv initialization;
LoadMLModels;
Mqtt initialization;
Mqtt.connect(Room);
sourcevideo = ClickEvent(e).video;
while ML == on do
    timeStart = time.now();
    iteration++; captureIMG = OpencvVideoCapture(sourceVideo);
    resizeIMG = OpencvImageResize(CaptureIMG,96,96);
    grayIMG = OpencvColor(resizeIMG,RGBA2GRAY);
    face = faceDetectModel.detectMultiScale(grayIMG);
    if face.exists then
        if MobilenetML == on then
            tensor = createTensor(48,48);
            logits = mobilenetModel.infer(tensor,convolution);
            predict(logits,face.resize(48,48));
            drawResults();
        end
        if KerasML == on then
            faceIntegers = UInt8ClampedArray(face);
            serializedData = imageByteArray(faceIntegers.resize(48,48));
            kerasModel.predict(serializedData);
            drawResults();
        end
        if OpenVinoML == on then
            message['image'] = base64(captureIMG);
            message['iteration'] = iteration;
            message['Room'] = Room;
            Mqtt.sendMessage(message);
            async drawResults();
        end
    else
        draw(captureIMG);
    end
    delay = 1000/framerate - (timeStart.now() - timeStart);
    setTimeout(delay);
end

```

1

Σχήμα 7.4: Κεντρικός βρόγχος μηχανικής μάθησης

Στο σχήμα υπάρχει ο κεντρικός βρόγχος της μηχανικής μάθησης. Για κάθε επανάληψη καλείται μια καθυστέρηση ίση με την μεταβλητή "delay" εάν ο χρόνος εκκίνησης και η καθυστέρηση είναι μικρότερα από τον τωρινό χρόνο. Εάν τα αντίστοιχα κουμπιά είναι σε

λειτουργία "on", τότε τρέχει ο κώδικας που είναι μέσα στις δηλώσεις "if".

7.4 Περιγραφή κώδικα στον διακομιστή Ubuntu με το MQTT

Στον διακομιστή τρέχει ένας εξυπηρετητής που μεταδίδει και λαμβάνει μηνύματα MQTT. Η υλοποίηση, γίνεται με την βιβλιοθήκη Paho. Όταν ένας συνδρομητής αιτείται σύνδεση, το Paho, διατρέχει τα στοιχεία του συνδρομητή και αν είναι έγκυρα τον συνδέει. Μετά ο συνδρομητής μπορεί να στείλει μήνυμα ως εκφωνητής σε όλους τους χρήστες του δωματίου που έγινε συνδρομητής. Αυτό είναι μια προσέγγιση που έγινε για να μην υπάρχουν πολλαπλές ίδιες αναλύσεις του ίδιου χρήστη.

Εφόσον ο εκφωνητής στείλει την εικόνα, τότε διατρέχεται μια διεργασία που εκτελεί την μηχανική μάθηση μέσω του OpenVino. Όταν επιστρέψει η διεργασία, το πρόγραμμα διαβάζει τα αποτελέσματά της και τα διαμορφώνει κατάλληλα για χρήση σε JSON μορφή. Τα αποτελέσματα της μηχανικής μάθησης, καθώς και τα υπόλοιπα στοιχεία που έστειλε ο εκφωνητής, μεταδίδονται πίσω προς όλους τους συνδρομητές του καναλιού.

Ο κώδικας βρίσκεται στον σύνδεσμο: "<https://github.com/mikePaps/master-thesis-papadakisM.git>"

8 Μοντέλα Μηχανικής Μάθησης σε πραγματικό χρόνο

8.0.1 Πολλαπλά συνελικτικά δίκτυα με Keras

Αρχικά προετοιμάζεται το μοντέλο μηχανικής μάθησης από Keras σε TensorFlow.

```
TensorFlowjs\_converter --input\_format keras \
                        path/to/my\_model.h5 \
                        path/to/tfjs\_target\_dir
```

1. Δημιουργείται ένας πίνακα ακεραίων με τα δεδομένα του προσώπου με τιμές εύρους από 0 έως 255. Αν η τιμή δεν είναι ακέραιος ή ξεπερνά τα όρια, ο πιο κοντινός αριθμός στο εύρος επιλέγεται.
2. Από τον πίνακα που δημιουργείται, εξάγονται τα pixels σε έναν πίνακα μίας διάστασης για 4 κανάλια (επειδή η εικόνα που διαβάζει το OpenCV έχει 4 κανάλια κόκκινο,πράσινο,μπλε και την τιμή άλφα).
3. Κατασκευάζεται μια εικόνα μεγέθους 48*48 pixels (ορισμένες από το μοντέλο) με βάση τον γραμμικό πίνακα του προηγούμενου βήματος.
4. Δημιουργείται ένας τένσορας με βάση την εικόνα, επιπέδου 1.
5. Ύστερα αφαιρείται η μέση τιμή από όλα τα εικονοστοχεία (με βήμα στήλης) και μετά διαιρούνται με την μέση απόσταση των στοιχείων από την προαναφερομένη μέση τιμή.
6. Κανονικοποιείται σε τένσορα [1, ύψος, πλάτος, κανάλια] και αποθηκεύεται σε ξεχωριστή μεταβλητή που επιστρέφει την κανονικοποιημένη τιμή.
7. Γίνεται καθαρισμός του τένσορα από την μνήμη, γιατί το TensorFlow, δεν καθαρίζει μόνο του τους τένσορες, που δεν υπάρχει πια αναφορά και γεμίζει η μνήμη.
8. Γίνεται πρόβλεψη με το μοντέλο, περιμένει το πρόγραμμα την απάντηση και επιστρέφονται τα αποτελέσματα.
9. Γίνεται τροποποίηση των δεδομένων για να μπορέσει να γίνει η σύγκριση των

αποτελεσμάτων με την συνάρτηση μείωσης, για λόγους ταχύτητας.

10. Το αποτέλεσμα της συνάρτησης μείωσης είναι και το τελικό συναίσθημα που πρόβλεψε η μηχανική μάθηση.

8.0.2 Πολλαπλά συνελικτικά δίκτυα με OpenVino

Στο πρόγραμμα της Intel, (στο αρχείο `classification_sample/classification_sample.py`) έγινε η επέκταση της εκτύπωσης των αποτελεσμάτων με την βιβλιοθήκη `pickle.dumps`.

```
dataToSend[str(det_label)] = probs[id]
print(pickle.dumps(str(dataToSend)))
```

Επειδή στον διακομιστή του OpenVino, καλείται η πρόβλεψη του νευρωνικού δικτύου μέσω παιδιού-διεργασίας και η απάντηση λαμβάνεται από την μητέρα διεργασία, υπάρχει πρόβλημα στην κωδικοποίηση του κειμένου. Έτσι μετά από την επιστροφή του αποτελέσματος, αυτό επαναφέρεται σε αντικείμενο με κωδικοποίηση χαρακτήρων UTF-8.

```
process = Popen(["python3.6", "/opt/intel/openvino_2019.1.144/
inference_engine/samples/python_samples/classification_sample/
classification_sample.py", "-i", msg['img'], "-m",
"/opt/intel/openvino_2019.1.144/inference_engine/samples/
python_samples/classification_sample/int8/
emotions-recognition-retail-0003.xml"], stdout=PIPE)

(output, err) = process.communicate()
exit_code = process.wait()
output = output.decode("utf-8")
```

8.1 Δημιουργία Συνάρτησης μηχανικής μάθησης με Mobilenet

Για την εκπαίδευση του πρώτου μοντέλου χρησιμοποιήθηκε το TensorFlow API (Javascript). Για την εκπαίδευση και αξιολόγηση εφαρμόζεται το σύνολο δεδομένων Fe2013. Η συμπίεσμένη έκδοση του συνόλου δεδομένων είναι 92 MB, ενώ η μη συμπίεσμένη έκδοση

είναι 295 MB. Υπάρχουν 28.000 εικόνες για εκπαίδευση και 3.000 για δοκιμές στο σύνολο δεδομένων. Κάθε εικόνα είναι αποθηκευμένη ως 48×48 εικονοστοιχεία. Το καθαρό σύνολο δεδομένων αποτελείται από τα εικονοστοιχεία της εικόνας, τα συναισθήματα που εκφράζουν και το σετ στο οποίο ανήκουν.

8.1.1 Κανονικοποίηση των δεδομένων

Για την βέλτιστη εκπαίδευση του αλγορίθμου πρέπει να δεδομένα να είναι κανονικοποιημένα. Δηλαδή η κάθε εικόνα να έχει ίδιες διαστάσεις με την προηγούμενη και ίδιο αριθμό καναλιών. Έτσι όταν οι εικόνες είναι μεγαλύτερες ή έχουν άλλες διαστάσεις, αποκόπτονται ή στρεβλώνονται και τα κανάλια πολλαπλασιάζονται με πίνακες φίλτρα.

8.1.2 Ορισμός του Συνελικτικό Νευρωνικό Δίκτυο

Ένα Συνελικτικό Νευρωνικό Δίκτυο χρησιμοποιεί φίλτρα στο ακατέργαστο εικονοστοιχείο μιας εικόνας για να μάθει το πρότυπο λεπτομερειών, σε σύγκριση με το γενικό πρότυπο με ένα παραδοσιακό νευρωνικό δίκτυο. Για να κατασκευαστεί ένα Συνελικτικό Νευρωνικό Δίκτυο, πρέπει να οριστεί:

- Ένα συνελικτικό επίπεδο
- Ένα επίπεδο συγκέντρωσης
- Πλήρως συνδεδεμένα επίπεδα

8.1.3 Επίπεδο Εισαγωγής εικόνων

Πρέπει να οριστεί ένας τένσορας ταυτόσημος με το σχήμα των δεδομένων. Μια εικόνα έχει ύψος, πλάτος και κανάλι. Το σύνολο δεδομένων του Kaggle Dataset είναι μια Μονοχρωματική εικόνα με μέγεθος 48×48 , ο τένσορας πρέπει να είναι μονοδιάστατος με μέγεθος 48×48 . Η εκπαίδευση του μοντέλου και η δοκιμή για την ευστοχία του, έγιναν σε περιβάλλον Nodejs με OpenCV.js και TensorFlow.js για Nodejs

8.1.4 Επίπεδο συνέλιξης

Επειδή η εφαρμογή πρέπει να υποστηρίζει άμεση επικοινωνία, χρειάζεται μια αρχιτεκτονική συνέλιξης που να χρειάζεται όσο το δυνατόν λιγότερη υπολογιστική ισχύ,

να βγάξει γρήγορα και σωστά αποτελέσματα. Ό,τι χρησιμοποιηθεί σε αυτό το βήμα, θα πρέπει να τρέξει και στην ζωντανή εφαρμογή, σε επεξεργαστές κινητών τηλεφώνων, κατά την διάρκεια ζωντανής συνεδρίας. Έτσι το Mobilenet, ικανοποιεί τις προδιαγραφές, καθώς με transfer learning, δύναται να παράξει γρήγορα και εύστοχα αποτελέσματα.

8.1.5 Επίπεδο Πρόβλεψης

Από το προηγούμενο βήμα βγαίνει ο πίνακας προβλέψεων, μέσα από την αρχιτεκτονική Mobilenet.

8.1.6 KNN - Ταξινομητής

Για να βρεθούν οι αποστάσεις μεταξύ των δεδομένων που έχουν περάσει μέσα από τον αλγόριθμο συνέλιξης και του μοντέλου που έχει δημιουργηθεί, πρέπει να υπάρχει ο τρόπος που ορίζει την απόσταση. Έτσι χρησιμοποιείται ο K-Nearest Neighbours classifier, για να οριστούν οι αποστάσεις των δεδομένων από το εκπαιδευμένο μοντέλο και είναι και ο τρόπος για να αποθηκευτεί ο ταξινομητής, για μετέπειτα χρήση.

9 Πειράματα και αποτελέσματα

Αρχικά για τα πειράματα χρειάστηκε να γίνει βαθμονόμηση, για την ικανότητα των διαθέσιμων πόρων να αντεπεξέλθουν στις απαιτήσεις του συστήματος. Ο κεντρικός πυρήνας της εφαρμογής είναι η διαδικασία εκπομπής και παρακολούθησης ζωντανής μετάδοσης βίντεο μεταξύ των χρηστών της. Οπότε οι αρχικές συνθήκες που πρέπει να ικανοποιούν τα μηχανήματα των χρηστών είναι:

1. Η σύνδεσή τους επιτυχώς στην εφαρμογή
2. Η χρήση ενός σύγχρονου περιηγητή ιστού, που έχει υλοποιημένες τις τεχνολογίες WebRTC
3. Να έχουν κάμερα και μικρόφωνο
4. Επαρκές εύρος ζώνης δικτύου για την ζωντανή επικοινωνία

Η μηχανική μάθηση είναι μια πράξη με μεγάλη υπολογιστική πολυπλοκότητα. Επιπλέον τα μοντέλα και οι βιβλιοθήκες που έχουν οριστεί, λειτουργούν με βάση την κεντρική μονάδα επεξεργασίας και δεν είναι παραλληλοποιήσιμα. Οπότε όσο πιο δυνατή είναι η κεντρική μονάδα επεξεργασίας του μηχανήματος, που εκτελούνται τα νευρωνικά δίκτυα, τόσο πιο πολλές επαναλήψεις-προβλέψεις μπορούν να γίνουν κάθε δευτερόλεπτο. Δύο είναι οι δείκτες απόδοσης της εφαρμογής:

1. Εάν μπορεί να αναγνωρίσει και να εφαρμόσει την μηχανική μάθηση πάνω στην επιλεγμένη μετάδοση βίντεο και να αποδώσει αποτέλεσμα (που είναι το συναίσθημα του προσώπου που αναγνωρίζει το βίντεο)
2. Πόσες εικόνες μπορεί να αναγνωρίσει και να επεξεργαστεί το δευτερόλεπτο.

Σε όλα τα πειράματα, τα μοντέλα μηχανικής μάθησης δούλεψαν επιτυχώς. Η ευστοχία τους καθορίζεται από το θεωρητικό μοντέλο που ακολουθούν και τις συνθήκες του βίντεο. Για να μελετηθεί η απόδοση και η συμπεριφορά κάθε μηχανήματος, πρέπει οι πράξεις του αλγορίθμου να αποσαφηνιστούν και να μετρηθεί ο χρόνος χρειάζονται για την διεκπεραίωσή τους. Πάντα το πρώτο βήμα του αλγορίθμου είναι να εντοπίσει το πρόσωπο του χρήστη στο βίντεο προς ανάλυση. Τα επιπλέον βήματα χωρίζονται σε δύο κατηγορίες:

1. Εάν ο χρήστης επιλέξει να χρησιμοποιήσει το μοντέλο μηχανικής μάθησης που

υπάρχει στον εξυπηρετητή (OpenVino).

2. Εάν ο χρήστης επιλέξει να χρησιμοποιήσει το μοντέλο μηχανικής μάθησης τοπικά στον υπολογιστή του (Keras).

Εάν ο χρήστης επιλέξει να χρησιμοποιήσει OpenVino, οι μετρικές πέραν της αναγνώρισης προσώπου είναι ο χρόνος που έκανε το στιγμιότυπο της φωτογραφίας του χρήστη να πάει στον εξυπηρετητή, αυτός να το επεξεργαστεί και να επιστρέψει το αποτέλεσμα στον χρήστη. Αυτό ονομάζεται RTT (Round Trip Time). Εάν ο χρήστης επιλέξει να χρησιμοποιήσει Keras, σε κάθε επανάληψη του αλγορίθμου ενεργοποιείται η ανάλυση μηχανικής μάθησης στο τερματικό του χρήστη. Αυτό σημαίνει πως επιβαρύνεται η ίδια η συσκευή και προκαλεί την δυσλειτουργικότητα της, όταν έχει παραπάνω πράξεις από όσες μπορεί να επεξεργαστεί. Η εφαρμογή έχει σχεδιαστεί έτσι ώστε να μην επηρεάζεται αρνητικά η εμπειρία χρήστη στο μηχάνημά του, ακόμα και όταν ο αλγόριθμος μηχανικής μάθησης καταναλώνει πολλούς πόρους. Τα παρακάτω πειράματα έχουν γίνει με γνώμονα την μέγιστη ισχύ που μπορεί να αποδώσει το κάθε μηχάνημα, ύστερα από την βαθμονόμηση τους. Τα πειράματα έγιναν πάνω σε τρεις συσκευές:

1. Ένα Laptop Asus VivoBook S14, με επεξεργαστή Intel Core i7-8565U @ 1.80GHz και μνήμη 16GB RAM.
2. Ένα Android Smartphone Samsung A40, με επεξεργαστή Exynos 7 Octa 7904 Octa core Processor και μνήμη 4GB RAM.
3. Ένα Android Smartphone Xiaomi Mi A1, με επεξεργαστή Qualcomm Snapdragon 625 2.0GHz octa-core Cortex A53 14nm FinFET και μνήμη 4GB RAM.
4. Ο εξυπηρετητής του OpenVino έχει 32GB RAM και 4 CPU (4 threads), τύπου Intel i7 3770 (2012).

9.1 Πείραμα με το λάπτοπ Asus VivoBook

9.1.1 Πείραμα με χρήση OpenVino

Το πείραμα έτρεξε για 49 επαναλήψεις και μέχρι 4 εικόνες το δευτερόλεπτο.

Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	14,595/49	185	478	297.85	275
RTT	52,295	483	2,571	1,067.25	1,007
Total	66,890	936	3,049	1,365.10	1,282
Server Time	33,041	473	712	674	679
Communication Time	4,793	62	491-timeout	98.72	65

Πίνακας 9.1: Πίνακας αποτελεσμάτων με λάπτοπ, OpenVino

Από το πείραμα εξάγουμε ότι στην καλύτερη περίπτωση, όπου ούτε ο εξυπηρετητής, ούτε ο υπολογιστής είναι υπερφορτωμένος, μπορεί να γίνει αποτελεσματική ανάλυση ενός στιγμιότυπου το δευτερόλεπτο. Εάν υπερφορτωθεί με παραπάνω εικόνες ο εξυπηρετητής ή ο υπολογιστής του χρήστη, τότε κάποιος από τους δύο σταματά και να ανταποκρίνεται, οπότε υπάρχει και απώλεια δεδομένων, διότι δεν επαρκεί η υπολογιστική ισχύς για να επεξεργαστεί την εικόνα και να λειτουργήσει κανονικά η επικοινωνία.

9.1.2 Πείραμα με χρήση Keras

Πείραμα με Keras, 140 επαναλήψεις, μέχρι 4 εικόνες το δευτερόλεπτο

Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	28,247/140	107	420	201.76	184
Keras	4,070	9	2,080	29.07	12
Total	32,317	116	2,500	230.83	196
Server Time	-	-	-	-	-
Communication Time	-	-	-	-	-

Πίνακας 9.2: Πίνακας αποτελεσμάτων με λάπτοπ, Keras

Εδώ φαίνεται ότι τοπικά οι χρόνοι είναι πολύ καλύτεροι. Ο ελάχιστος χρόνος είναι 116 ms, που είναι πολλά υποσχόμενος, και ο Διάμεσος 196. Οι μέγιστοι χρόνοι εμφανίζονται κατά την εκκίνηση της διαδικασίας, διότι εκεί φορτώνονται οι βιβλιοθήκες, οπότε δεν επηρεάζουν το αποτέλεσμα σε βάθος χρόνου.

9.2 Πείραμα με το smartphone Xiaomi Mi A1

9.2.1 Πείραμα με χρήση OpenVino

Το πείραμα έτρεξε για 101 επαναλήψεις, μέχρι 4 εικόνες το δευτερόλεπτο.

Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	59,926/101	569	631	593.32	592
RTT	155,348	1,214	1,632	1,538.1	1,286
Total	215,274	1,783	2,563	2,131.42	1,878
Server Time	65,448	372	732	648	652
Communication Time	4793	62	491-timeout	98.72	65

Πίνακας 9.3: Πίνακας αποτελεσμάτων με smartphone Xiaomi Mi A1, OpenVino

Η υλοποίηση στο κινητό είναι αρκετά πιο δύσβατη. Επειδή δεν έχουν μεγάλη υπολογιστική ισχύ πολλές φορές στέλνουν το αίτημα για επεξεργασία στον εξυπηρετητή, αλλά είναι απασχολημένο και θα χάσει την απάντηση. Ο ολικός χρόνος επιτρέπει 1 φωτογραφία το 1.8 δευτερόλεπτα.

9.2.2 Πείραμα με χρήση Keras

Πείραμα με Keras, 63 επαναλήψεις, μέχρι 4 εικόνες το δευτερόλεπτο.

Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	37,307/63	549	659	601.75	601
Keras	5,557	65	866	89.6	76
Total	42,864	614	1,525	691.35	677
Server Time	-	-	-	-	-
Communication Time	-	-	-	-	-

Πίνακας 9.4: Πίνακας αποτελεσμάτων με smartphone Xiaomi Mi A1, Keras

Αντίστοιχα φαίνεται ότι όταν δεν υπάρχει επικοινωνία με εξυπηρετητή, το πρόγραμμα δουλεύει πιο άρτια. Οι χρόνοι είναι πιο μικροί και επιτρέπουν ανάλυση κάτι παραπάνω από μία εικόνα το δευτερόλεπτο.

9.3 Πείραμα με το smartphone Samsung Galaxy A40

9.3.1 Πείραμα με χρήση OpenVino

Το πείραμα έτρεξε για 106 επαναλήψεις, μέχρι 4 εικόνες το δευτερόλεπτο.

Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	50,123/106	426	515	472.85	474
RTT	146,680	955	1,619	1,383.78	1,567
Total	196,803	1,381	2,134	1,856.63	2,041
Server Time	66,878	485	718	630.92	632
Communication Time	4,793	62	491-timeout	98.72	65

Πίνακας 9.5: Πίνακας αποτελεσμάτων με smartphone Samsung Galaxy A40, OpenVino

Σε ένα πιο σύγχρονο και ισχυρό κινητό από το προηγούμενο, υπάρχει την αλλαγή στους χρόνους. Μια εικόνα μπορεί να αναλυθεί σε μισό δεύτερο πιο γρήγορα από την προηγούμενη συσκευή. Παρόλα αυτά, όταν η συσκευή είναι υπό πίεση, πάλι αποτυγχάνει να λάβει όλα τα μηνύματα του εξυπηρετητή.

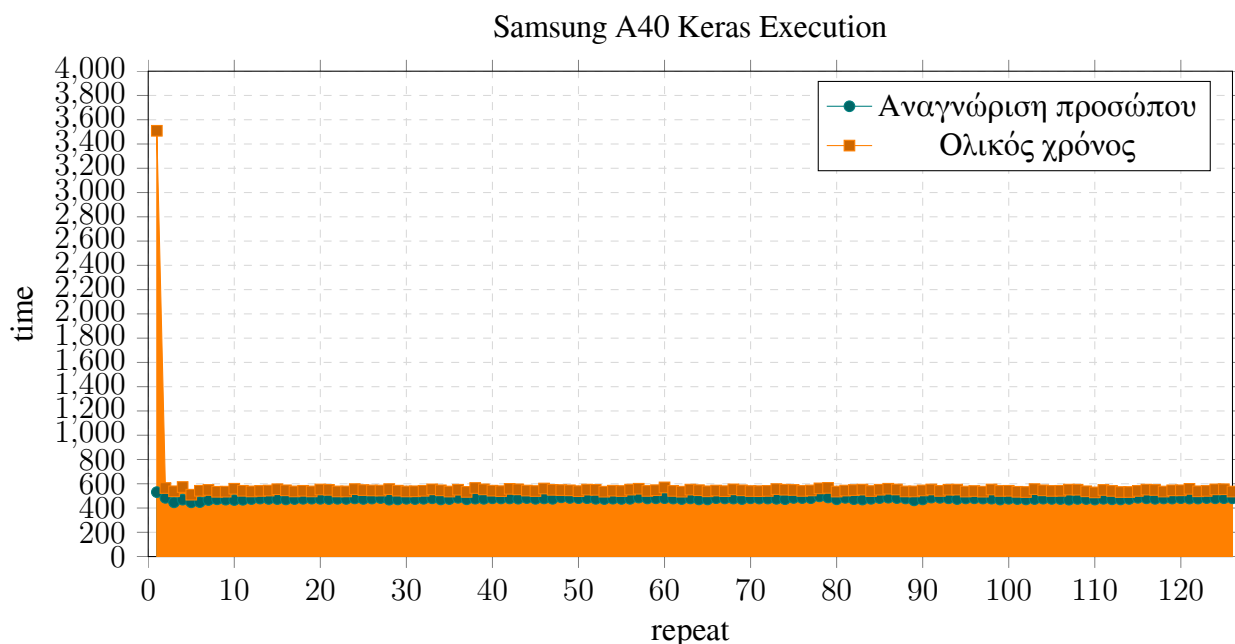
9.3.2 Πείραμα με χρήση Keras

Πείραμα με Keras, 63 επαναλήψεις, μέχρι 4 εικόνες το δευτερόλεπτο.

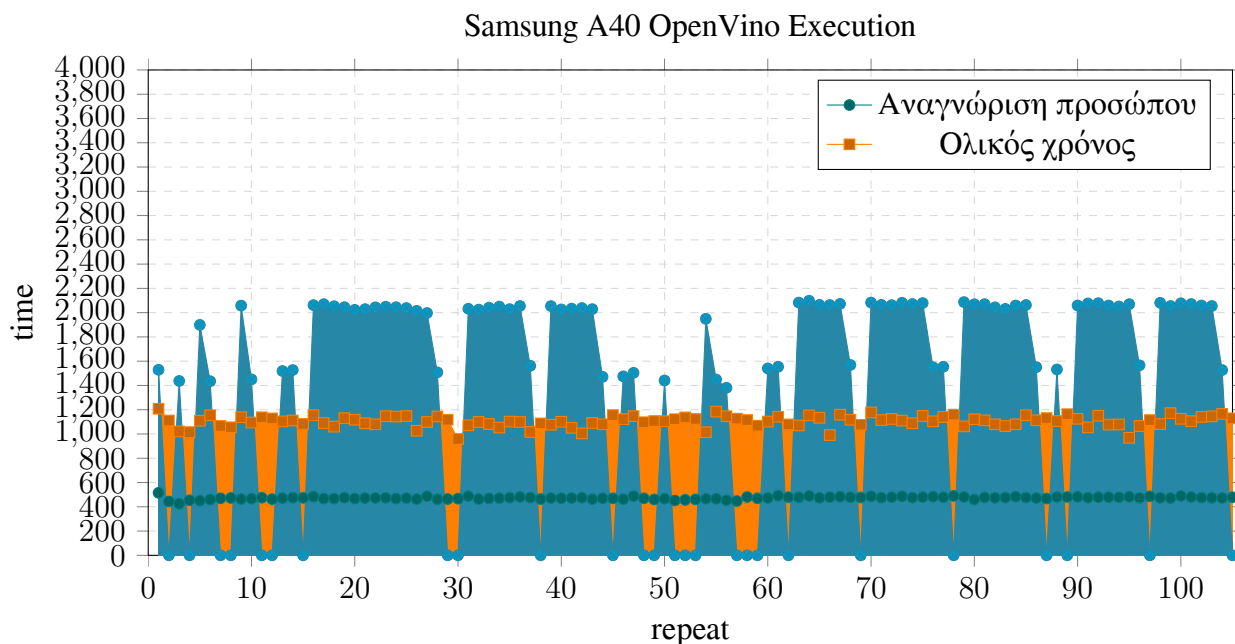
Μέθοδος\Χρόνος (ms)	Ολικός	Ελάχιστος	Μέγιστος	Μέσος	Διάμεσος
Face Detection	60,995/128	447	531	476.52	476
Keras	11,764	63	2978	91.9	68
Total	72,759	510	3509	568.42	544
Server Time	-	-	-	-	-
Communication Time	-	-	-	-	-

Πίνακας 9.6: Πίνακας αποτελεσμάτων με smartphone Samsung Galaxy A40, Keras

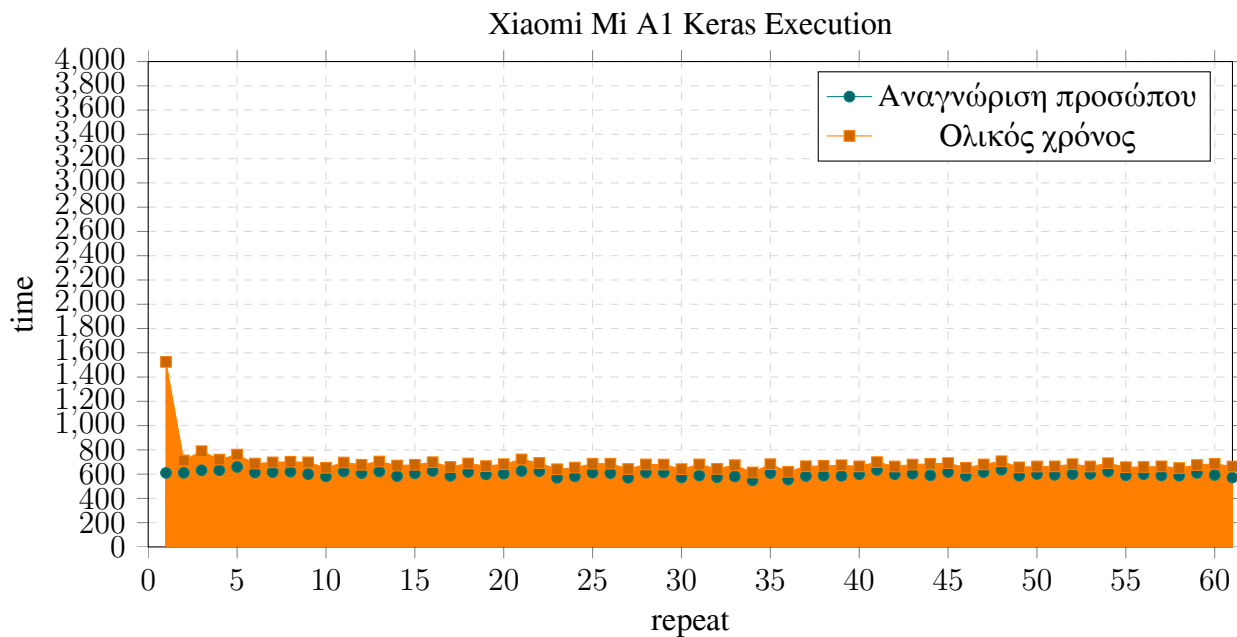
Σε αυτό το πείραμα φαίνεται ότι οριακά δεν επιτυγχάνεται δύο εικόνες το δευτερόλεπτο. Οι χρόνοι είναι πολύ σταθεροί και η ανάλυση γίνεται σε περίπου 0.5 δευτερόλεπτα.



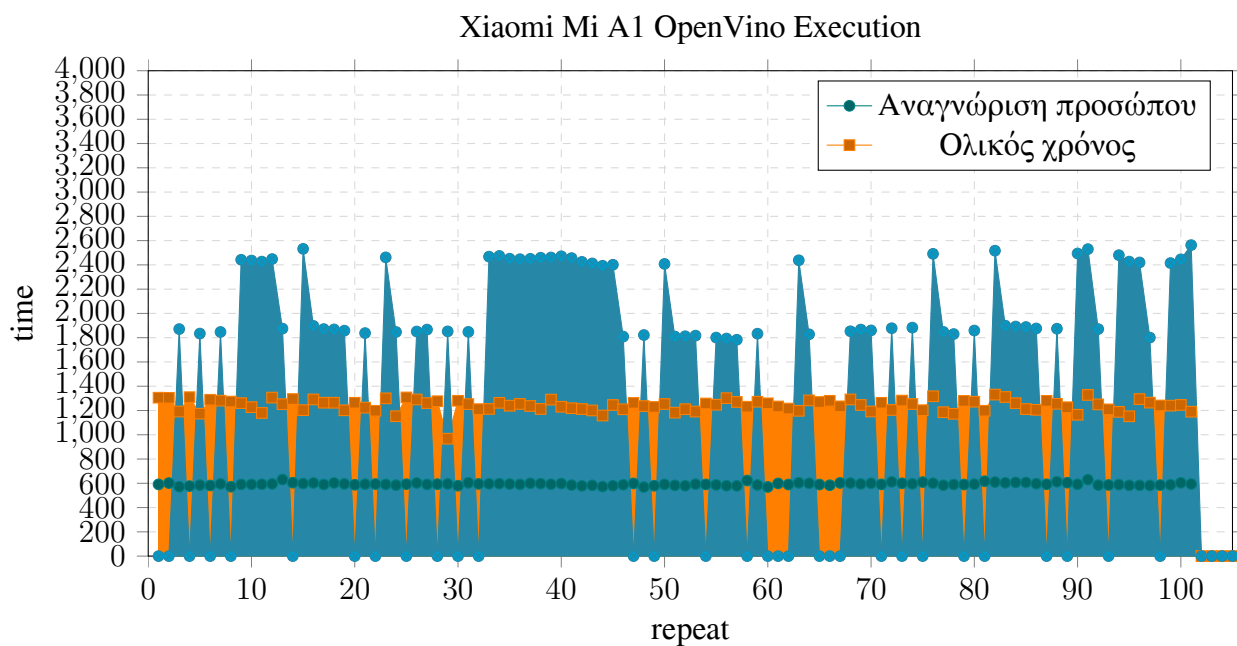
Από αυτό το γράφημα φαίνεται πως στον υπολογιστή, μέσα σε ένα δευτερόλεπτο, σχεδόν οριακά δεν χωράει η επανάληψη. Το μεγαλύτερο κομμάτι της εκτέλεσης το καταλαμβάνει ο αλγόριθμος που ανιχνεύει το πρόσωπο.



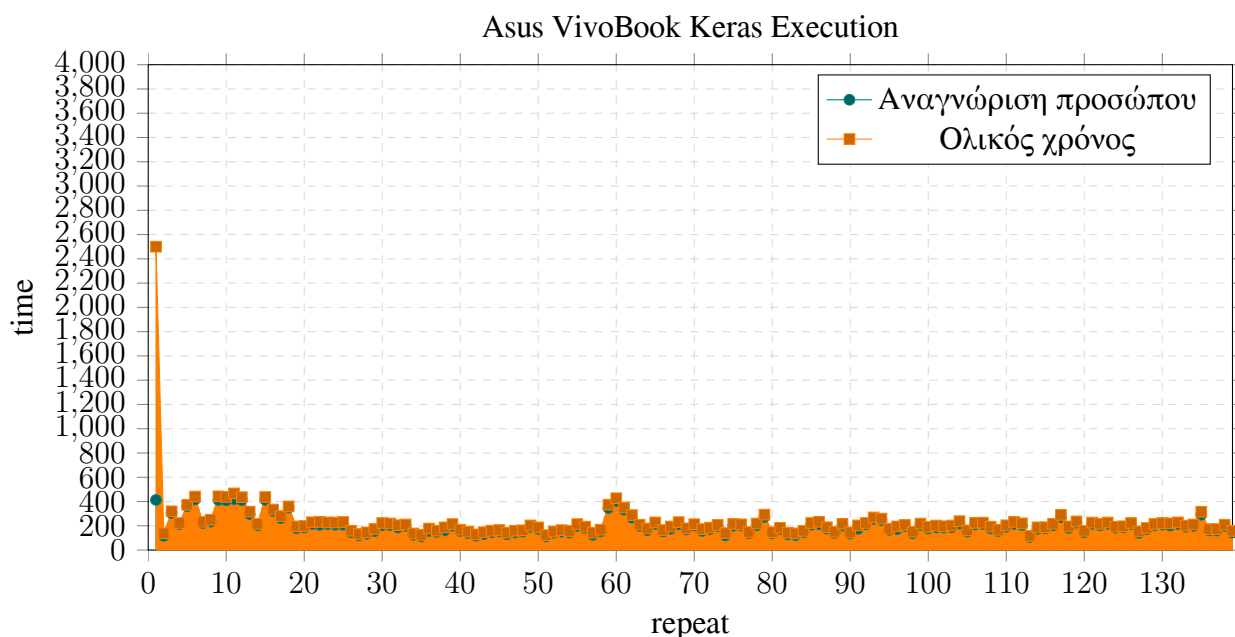
Εδώ διαπιστώνεται η δυσκολία του κινητού να επικοινωνήσει με τον εξυπηρετητή όταν αυτό είναι απασχολημένο με την εύρεση του προσώπου. Οι χρόνοι επικοινωνίας ενώ θα έπρεπε να είναι κοντά στα 62+62 ms, είναι αρκετά υψηλότεροι και υπάρχουν μεγάλες απώλειες πακέτων. Ως εκ τούτου, παράγει αποτελέσματα πολύ αργότερα από όσο περιμένει ο χρήστης.



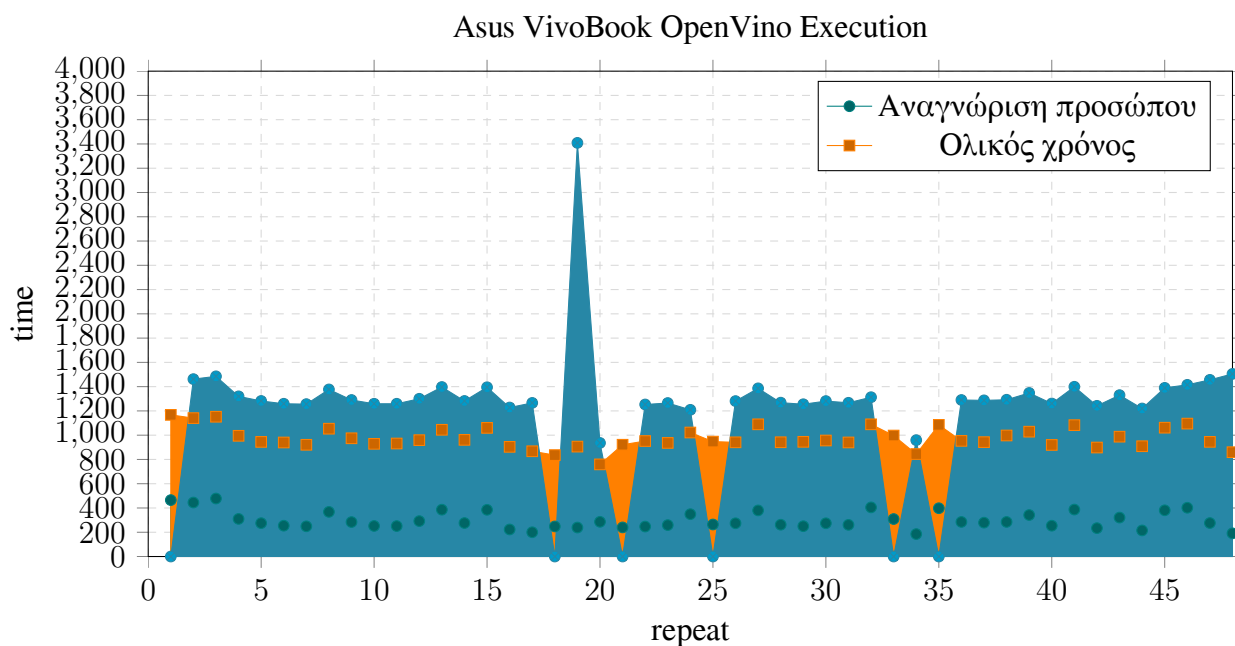
Ο ολικός χρόνος όπως και πριν επηρεάζεται κυρίως από την εύρεση του προσώπου στο περιβάλλον. Εδώ λόγω του μοντέλου του κινητού ο χρόνος πλησιάζει το 1 δευτερόλεπτο. Οπότε η εφαρμογή δεν είναι ιδιαίτερα ανταποκρίσιμη.



Όπως και πριν, το κινητό χάνει πολλά δεδομένα που του στέλνει ο εξυπηρετητής. Για τον εξυπηρετητή ο ρυθμός που στέλνει το κινητό είναι εντός του πλαισίου του και δεν έχει καμία απώλεια. Όμως επειδή το κινητό δεν έχει, συγκριτικά, καλή υπολογιστική ισχύ, το μέγεθος των δεδομένων που χάνονται είναι αρκετά υψηλό.



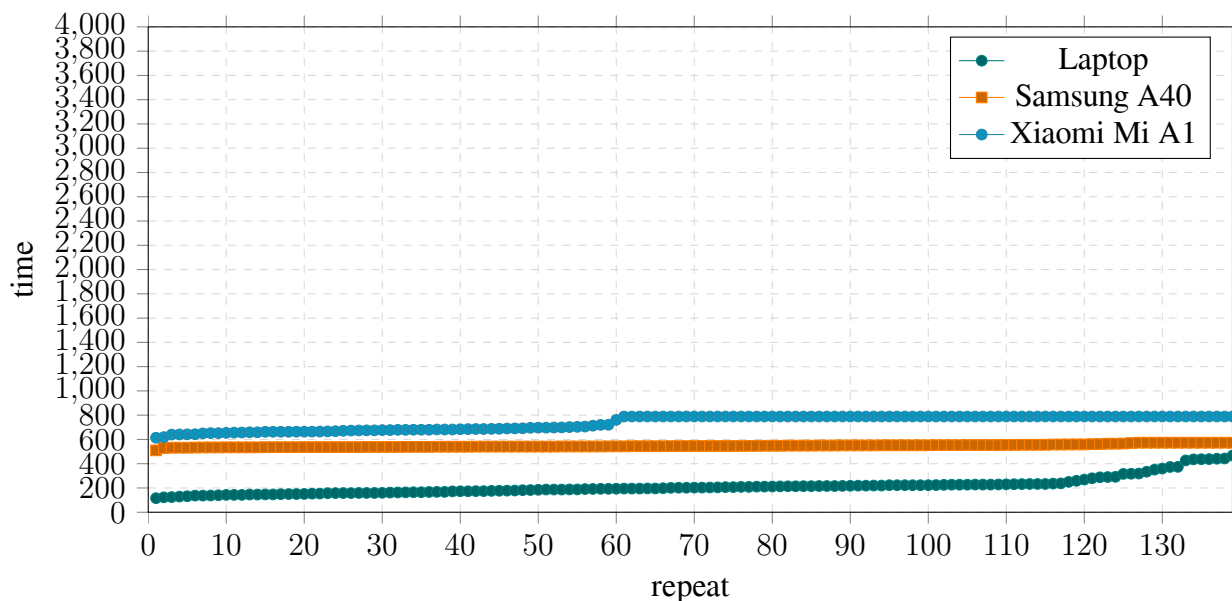
Το Laptop, έχοντας πολύ ισχυρή κεντρική μονάδα επεξεργασίας, μπορεί να κάνει επεξεργασία πολλών εικόνων το δευτερόλεπτο. Η μηχανική μάθηση που βρίσκει το πρόσωπο είναι σχεδόν εξ ολοκλήρου ο παράγοντας που επηρεάζει το αποτέλεσμα.



Εφόσον ο υπολογιστής μπορεί να κάνει επεξεργασία πάνω από 4 εικόνες το δευτερόλεπτο, ανταποκρίνεται κανονικά στην συνομιλία του με τον εξυπηρετητή. Αλλά για τον εξυπηρετητή 4 εικόνες το δευτερόλεπτο είναι οριακό. Έτσι λόγω φόρτου, δεν μπορεί να δεχτεί όλα τα μηνύματα του χρήστη και έτσι κάποιες φορές είναι απασχολημένο και χάνονται.

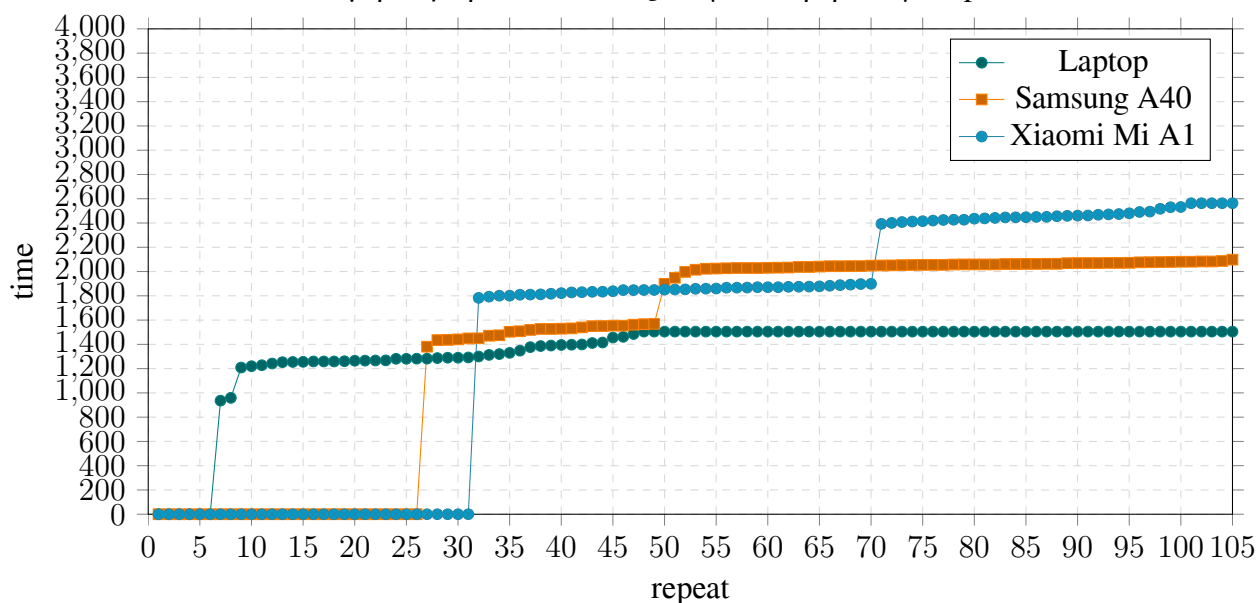
Τα επόμενα δύο διαγράμματα έχουν ως σκοπό την ανάδειξη μιας αναπαράστασης υπολογιστικής δύναμης, φόρτου και χρόνου εκτέλεσης.

Σύγκριση Χρόνου και αυξανόμενου φόρτου με Keras



Στο διάγραμμα φαίνεται η διαβάθμιση της δυσκολίας του προγράμματος καθώς το laptop είναι πιο γρήγορο από τα κινητά, αλλά η διακύμανση είναι αρκετά μεγάλη. Από 100 ms έως 500 ms. Τα κινητά έχουν υψηλότερο χρόνο εκτέλεσης, αλλά η διακύμανσή τους είναι πιο σταθερή. Αυτή εξαρτάται από πόσες διεργασίες είναι στο φόντο και τι κάνουν αυτές οι διεργασίες.

Σύγκριση Χρόνου και αυξανόμενου φόρτου με OpenVino



Αρχικά φαίνονται οι φορές που δεν έλαβε γνώση ο χρήστης για το αποτέλεσμα της

ανάλυσης. Δηλαδή ενώ έστειλε στον εξυπηρετητή αίτημα για ανάλυση συναισθήματος, δεν πήρε ποτέ την απάντηση. Αυτό συμβαίνει διότι είτε ο εξυπηρετητής είτε ο πελάτης έχουν τόσο φόρτο που χάνουν πακέτα επικοινωνίας. Με τις ίδιες παραμέτρους, η χειρότερη περίπτωση υφίσταται στον χρήστη με τους λιγότερους πόρους. Ενώ η κατάσταση θα ήταν παρόμοια για τον χρήστη με το laptop, εάν έβαζε παραπάνω από 4 εικόνες το δευτερόλεπτο. Διότι όχι μόνο ο εξυπηρετητής θα είχε πρόβλημα στο να εξυπηρετήσει τόσες εικόνες, έτσι ώστε να κάνει ουρά, αλλά θα χάνει πολλά δεδομένα, επειδή δεν έχει διαθέσιμους πόρους, ώστε να λάβει καινούργια δεδομένα. Από την γραφική επιπλέον βγαίνει το συμπέρασμα ότι κλιμακωτά ο εξυπηρετητής μπορεί να καθυστερήσει να ανταποκριθεί ή τα κινητά, λόγω φόρτου, έχουν αργή ανταπόκριση.

10 Μελλοντικές Επεκτάσεις και μελέτες

Η εφαρμογή μπορεί να λειτουργήσει σε κεντριοποιημένα περιβάλλοντα. Ενδιαφέρον θα ήταν η εφαρμογή πολλαπλών συσκευών, αποκεντριοποιημένα ή ενός κατανεμημένου δικτύου, για να γίνεται εξισορρόπηση πόρων. Ενώ ένα σύστημα κοντά στο P2P έχει αξία όταν ο κάθε κόμβος μπορεί να διεκπεραιώσει την λειτουργία της ανάλυσης και του βίντεο, ειδικά καθώς βελτιώνονται οι αλγόριθμοι και οι τεχνολογίες, για τώρα φαίνεται να είναι δύσχερο όπως είναι η εφαρμογή.

Η εφαρμογή να υλοποιηθεί σαν native εφαρμογή στα smartphone. Οι βιβλιοθήκες στην Javascript που χρησιμοποιούνται μπορούν να αντικατασταθούν από native κώδικα. Έτσι υπάρχει καλύτερη πρόσβαση στους πόρους και μεγαλύτερη ευελιξία ως προς την παραλληλοποίηση των λειτουργιών.

Να χρησιμοποιηθεί η κάρτα γραφικών όπου είναι εφικτό. Έχοντας υλοποιήσει όλες τις βιβλιοθήκες σε γλώσσα προγραμματισμού για τον περιγητή ιστού, υπάρχουν δυνατότητες χρησιμοποίησης παράλληλης επεξεργασίας, μέσω της κάρτας γραφικών. Σίγουρα θα μείωνε εκπληκτικά τους χρόνους ανταπόκρισης. Να γίνει η ανάλυση της εικόνας εξ ολοκλήρου στο σύννεφο. Από τα αποτελέσματα φαίνεται ότι η ανάλυση εικόνας για την εύρεση προσώπου περιέχει το μεγάλο κόστος για τις συσκευές. Έτσι αν η βίντεο μετάδοση περνούσε μέσα από έναν εξυπηρετητή θα ήταν πιο γρήγορη η ανάλυση. Δεν υπάρχουν όμως κριτήρια για το πότε τελικά είναι επαρκής.

Βιβλιογραφία

- André, E., Le Breton, N., Lemesle, A., Roux, L., and Gouaillard, A. (2018). Comparative study of webrtc open source sfus for video conferencing. In *2018 Principles, Systems and Applications of IP Telecommunications (IPTComm)*, pages 1–8.
- Bouhabba, E. M., Shafie, A. A., and Akmeliawati, R. (2011). Support vector machine for face emotion detection on real time basis. In *2011 4th International Conference on Mechatronics (ICOM)*, pages 1–6.
- Dahmane, M. and Meunier, J. (2011). Emotion recognition using dynamic grid-based hog features. In *Face and Gesture 2011*, pages 884–888.
- Dan Duncan, Gautam Shine, C. E. (2016). Facial emotion recognition in real time.
- Devang C. Prajapati, Disha P. Vanani, K. H. T. K. R. P. N. P. B. (2015). Face detection in digital image: A technical review. *Disha P. Vanani et al Int. Journal of Engineering Research and Applications*, 5:83–88.
- Fernando, S., Scott-Brown, J., Şerban, O., Birch, D., Akroyd, D., Molina-Solana, M., Heinis, T., and Guo, Y. (2020). Open visualization environment (ove): A web framework for scalable rendering of data visualizations. *Future Generation Computer Systems*, 112:785 – 799.
- García, B., G. F. G. M. and Jiménez, E. (2018). User impersonation as a service in end-to-end testing. In *Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development*, pages 707–714.
- Gupta, S. (2018). Facial emotion recognition in real-time and static images. In *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, pages 553–560.
- Gwen Littlewort, Marian Stewart Bartlett, I. F. J. C. J. R. M. (2003). *Analysis of Machine Learning Methods for Real-Time Recognition of Facial Expressions from Video*. IJDET.
- Healy, M., Donovan, R., Walsh, P., and Zheng, H. (2018). A machine learning emotion detection platform to support affective well being. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2694–2700.
- Hou Lee, J. R. and Wong, A. (2020). Timeconvnets: A deep time windowed convolution neural network design for real-time video facial expression recognition. In *2020 17th Conference on Computer and Robot Vision (CRV)*, pages 9–16.
- Jadhav, D. and Bhosle, U. (2017). Surf based video summarization and its optimization. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 1252–1257.
- Keerthana Nagula, K Lakshmi Priya, K. G. S. R. S. (2019a). Analytical comparison of emotion using real time video and audio. *International Journal of Innovations in Engineering and Technology*, 13.
- Keerthana Nagula, K Lakshmi Priya, K. G. S. R. S. (2019b). Analytical comparison of emotion using real time video and audio. *International Journal of Innovations in Engineering and Technology*, 13.
- Khan, R. and Debnath, R. (2020). Human distraction detection from video stream using artificial emotional intelligence. *International Journal of Image, Graphics and Signal Processing*, 12:19–29.

- Leon, E., Clarke, G., Callaghan, V., and Sepulveda, F. (2007). A user-independent real-time emotion recognition system for software agents in domestic environments. *Engineering Applications of Artificial Intelligence*, 20(3):337 – 345.
- Li, Y., Cheng, H., Liu, Y., and Shen, C. (2019). Edge-computing convolutional neural network with homography-augmented data for facial emotion recognition. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3287–3291.
- Matlovic, T., Gaspar, P., Moro, R., Simko, J., and Bielikova, M. (2016). Emotions detection using facial expressions recognition and eeg. In *2016 11th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, pages 18–23.
- Michel, P. and El Kaliouby, R. (2003). Real time facial expression recognition in video using support vector machines. In *Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI '03*, page 258–264, New York, NY, USA. Association for Computing Machinery.
- Misra, A. (2020). Deep learning acceleration on the edge.
- Mukhopadhyay, S. and Sharma, S. (2020). Real time facial expression and emotion recognition using eigen faces, lbph and fisher algorithms. In *2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence)*, pages 212–220.
- MV. Ramana, N. Kavya Manjari, R. T. R. S. K. I. S. C. (2020). Analyzing real time human emotions using deep learning. *International Journal for Research in Applied Science & Engineering Technology*, 8.
- Nguyen Hoang, A. M. (2019). Computer vision deployment on edge devices.
- Norrgård, M. (2020). Using computer vision in retail analytics. 20.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.
- Rajesh, K. M. and Naveenkumar, M. (2016). A robust method for face recognition and face emotion detection system using support vector machines. In *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, pages 1–5.
- Rathod, P., George, K., and Shinde, N. (2016). Bio-signal based emotion detection device. In *2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 105–108.
- Rocha Neto, A. F., Delicato, F. C., Batista, T. V., and Pires, P. F. (2020). *Distributed Machine Learning for IoT Applications in the Fog*, chapter 12, pages 309–345. John Wiley & Sons, Ltd.
- Roman Hasko, Nataliya Shakhovska, O. V. R. H. (2020). *A Mixed Fog/Edge/AIoT/Robotics Education Approach based on Triple Learning*, volume 2616. CEUR-WS.
- Sinha, A. and R P, A. (2019). Real time facial emotion recognition using deep learning.
- Suk, M. and Prabhakaran, B. (2014). Real-time mobile facial expression recognition system - a case study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

- Vaibhav Mandavkar, M. S. S. K. (2020). Review of face recognition and face emotion detection system using open source computer vision (opencv). *CLIO An Annual Interdisciplinary Journal of History*, 6.
- Xianghua Fan, Fuyou Zhang, Haixia Wang, and Xiao Lu (2012). The system of face detection based on opencv. In *2012 24th Chinese Control and Decision Conference (CCDC)*, pages 648–651.