ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	(Ελληνικά) **Εφαρμογή και Ανάλυση στην αναγνώριση εικόνων με τη χρήση «Coral Tensor Processing Unit»** (Αγγλικά) **Performance Analysis of the accelerator "Coral Tensor Processing Unit" in Picture recognition**
Ονοματεπώνυμο Φοιτητή	Ιωάννης Φωτόπουλος
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	Π/ 18238
Επιβλέπων	Χρήστος Δουληγέρης, Καθηγητής

Ημερομηνία Παράδοσης Μήνας Έτος

2024/09

Περίληψη

Η Τεχνητή Νοημοσύνη είναι ένας από τους ταχύτερα αναπτυσσόμενους τομείς στην επιστήμη της Πληροφορικής. Το πρόβλημα που λύνει η συγκεκριμένη εργασία εντάσσεται στην περιοχή την σύγχρονων νευρωνικών δικτύων Τα σύγχρονα νευρωνικά δίκτυα διαχειρίζονται διεργασίες με μεγάλη πολυπλοκότητα και για αυτό απαιτούν μεγάλη υπολογιστική ισχύ. Οι σύγχρονοι επεξεργαστές γενικού σκοπού «δυσκολεύονται» να επιτύχουν επαρκείς και ικανοποιητικές αποδόσεις. Η Google ανέπτυξε τις Tensor Processing Units (TPUs) για να επιταχύνει την εκτέλεση εφαρμογών Τεχνητής Νοημοσύνης τόσο σε κέντρα δεδομένων όσο και σε λοιπές εφαρμογές, εκπληρώνοντας αυτό τον σκοπό. Σε αυτή τη πτυχιακή εργασία, ασχολούμαστε με τον Edge TPU. Το Edge TPU είναι ένα μικρό ολοκληρωμένο κύκλωμα που επιτρέπει την ανάπτυξη εφαρμογών ΤΝ “at the edge”. Το Edge TPU είναι ικανό να εκτελεί τέσσερα (4) τρισεκατομμύρια πράξεις ανά δευτερόλεπτο, χρησιμοποιώντας 2 Watt ισχύος. Ωστόσο, η αρχιτεκτονική και το σύνολο εντολών τέτοιων επιταχυντών ΤΝ, εμφανίζουν διάφορες προκλήσεις και περιορισμούς. Πραγματοποιήσαμε bench marking στο TPU, με έτοιμα μοντέλα που παρέχει η Google, με στόχο να αξιολογήσουμε τις δυνατότητές του. Τα αποτελέσματα που προέκυψαν αποκαλύπτουν σημαντική επιτάχυνση για το Google Edge TPU σε σύγκριση με τους επεξεργαστές BCM 2837 (Raspberry Pi 3, A+), AMD Ryzen 5 3500U. Συνολικά, επιτυγχάνεται σημαντική επιτάχυνση μεγάλου μεγέθους συνελικτικών νευρωνικών δικτύων και απλών τεχνητών νευρωνικών δικτύων. Το Edge TPU παρέχει έως και 10 φορές καλύτερη απόδοση από τον BCM 2837 και 7 φορές μεγαλύτερη απόδοση από τον AMD Ryzen 5 3500U.

Λέξεις κλειδιά: *Coral TPU, Νευρωνικό δίκτυο, Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Αναγνώριση αντικειμένων, Τεχνητά Νευρωνικά Δίκτυα, Σύνθετες Νευρωνικές Δομές, Αλγόριθμοι εκπαίδευσης, Επιταχυντές Μηχανικής Μάθησης, Tensor Flow Lite, Τεχνολογία TPU.*

Abstract

Artificial Intelligence is one of the fastest-growing fields in Computer Science. The problem that this work solves belongs to the area of modern neural networks. Modern neural networks manage complex processes and therefore require a lot of computing power. Modern general-purpose processors "struggle" to achieve adequate and satisfactory performance. Google developed Tensor Processing Units (TPUs) to accelerate the execution of AI applications in data centers and other applications, fulfilling this purpose. In this thesis, we deal with Edge TPU. The Edge TPU is a small integrated circuit that enables the development of IT applications "at the edge". The Edge TPU can perform four (4) trillion operations per second, using 2 Watts of power. However, the architecture and instruction set of such AI accelerators present various challenges and limitations. We benchmarked the TPU, with ready-made models provided by Google, to evaluate its capabilities. The obtained results reveal a significant speedup for Google Edge TPU compared to BCM 2837 (Raspberry Pi 3, A+), and AMD Ryzen 5 3500U processors. Overall, a significant speedup of large-scale convolutional neural networks and simple artificial neural networks is achieved. Edge TPU provides up to 10x better performance than BCM 2837 and 7x better performance than AMD Ryzen 5 3500U.

Keywords: *Coral TPU, Neural Network, Artificial Intelligence, Machine Learning, Object Recognition, Artificial Neural Networks, Complex Neural Structures, Training Algorithms, Machine Learning Accelerators, Tensor Flow Lite, TPU Technology.*

Περιεχόμενα

1. Εισαγωγή στα Νευρωνικά Δίκτυα και στις διεργασίες Τεχνητές Νοημοσύνης μέσω του επιταγχυντή «Coral Tensor Processing Units» ……7

1.1.Εισαγωγή……..……7

1.2.Η γενική κατεύθυνση της εργασίας ……7

1.3.Ιστορική Αναδρομή ……....7

1.4.Περιγραφή της λειτουργίας του TPU ..…….8

1.5.Συμπεράσματα…….9

2. Τα χαρακτηριστικά του TensorFlow Lite και η μεταφορά μάθησης ……..…10

2.1.Εισαγωγή……10

2.2.Τα τεχνικά χαρακτηριστικά του TensorFlow Lite ……10

2.3.Επισκόπηση συμβατότητας ……..11

2.4. Μεταφορά μάθησης …… ..……11

2.5.Διδακτική Προσέγγιση ……..

2.6.Συμπεράσματα……..12

3. Coral Tensor Processing Units ……..12

3.1.Εισαγωγή……...12

3.2.Η αρχιτεκτονική των Coral Tensor Processing Units ……12

3.3.Μεταγλώττιση …….. ……....12

3.4.Πλεονεκτήματα της χρήσης του Tensor Flow Lite ……..13

3.5.Συμπεράσματα……..13

4. Λογισμικά, εγκαταστάσεις, συγκρίσεις και μετρήσεις ……..14

4.1.Εισαγωγή……14

4.2.Η διαδικασία εγκατάστασης των προτεινόμενων λογισμικών ……14

4.3.Παραδείγματα και Bench Markings Windows ……..16

4.4.Π Αξιοποίηση του Raspberry PI ……18

4.5.Διδακτική Προσέγγιση ……21

4.6.Συμπεράσματα……..…21

Βιβλιογραφία – Δικτυογραφία ……22

Κατάλογος Εικόνων

Εικόνα 1 USB Accelerators …….……8
Εικόνα 2: Αρχιτεκτονικη Coral TPU ……..……9
Εικόνα 3: Αλγόριθμος μετατροπής για τον TensorFlow Lite ……10
Εικόνα 4: Μεταγλώτιση:…… ……12 Εικόνα 5: Διαδικασίες εγκατάστασης ……..14 Εικόνα 6: Εντολοδότηση…….……15
Εικόνα 7: Αποτελέσματα των διαδικασιών εγκατάστασης. ……..15
Εικόνα 8: Διαδικασίες εγκατάστασης …….. …..……16
Εικόνα 9: Κώδικας και οθόνη εργασίας: ……..……16
Εικόνα 10: Οθόνες εργασίας:.. ……..……17 Εικόνα 11: Οθόνες εργασίας για συγκρίσει λειτουργιών ……17
Εικόνα 12: Γράφημα αποτύπωσης απόδοσης λειτουργίας ……..19
Εικόνα 13: Κώδικας λειτουργίας …….……...20

1. Εισαγωγή στα Νευρωνικά Δίκτυα και σε διεργασίες Τεχνητές Νοημοσύνης μέσω του επιταγχυντή «Coral Tensor Processing Units»

1.1 Εισαγωγή

Η παρούσα πτυχιακή εργασία μελετά και περιγράφει μια αξιόπιστη μέθοδο για την αναγνώριση ειδών πουλιών, προσθέτοντας αξία στην έρευνα της τεχνητής νοημοσύνης και της μηχανικής μάθησης. Τα συμπεράσματα αφορούν τους δείκτες βελτίωσης μέσω της χρήσης Coral Edge Tensor Processing Units (TPUs), και συγκεκριμένα την βελτίωση της αποδοτικότητας και την ακρίβεια των συστημάτων αναγνώρισης εικόνας. Στο δεύτερο υποκεφάλαιο της πτυχιακής (1.2), καταθέτουμε την γενική της κατεύθυνση σχετικά με το θέμα της εργασίας αναφέροντας τον σκοπό, τους στόχους και τα προσδοκώμενα αποτελέσματα. Στο τρίτο υποκεφάλαιο (1.3), σημειώνουμε την ιστορική διαδρομή του θέματος σε σχέση με την ανάπτυξη της τεχνολογίας του TPU, σε συδυασμό με τις μεθόδους ακολουθήθηκαν στην έρευνα μας. Στο τέταρτο υποκεφάλαιο (1.4), περιγράφουμε την λειτουργία του USB Accelerator. Το 1ο Κεφάλαιο κλείνει με τις σχετικές βιβλιογραφικές αναφορές.

1.2 Η γενική κατεύθυνση της εργασίας

Το ερευνητικό πλαίσιο της παρούσας πτυχιακής εργασίας εντάσσεται στον τομέα μελετών σύγκρισης απόδοσης λογισμικών και πιο συγκεκριμένα, εντάσσεται στην συγκριτική μελέτη περιβαλλόντων τα οποία αξιοποιούν εργαλεία τεχνητής νοημοσύνης και μηχανικής μάθησης. Το εξειδικευμένο αντικείμενο της έρευνας εφαρμόζεται στην αναγνώριση ειδών πουλιών μέσω εικόνων. Η φύση της έρευνας αφορά την εφαρμογή των θεωρητικών γνώσεων στην πράξη, και εστιάζει στο πεδίο της ανασκόπησης των τεχνολογιών αναγνώρισης εικόνας. Το πρόβλημα αναφοράς είναι η βελτιστοποίηση της διαδικασίας αναγνώρισης ειδών πουλιών, μέσω μηχανικής μάθησης. Η έρευνα και η διακρίβωση των μεταβλητών της αξιοποιεί την πλατφόρμα PyCoral [1] και το Edge TPU [2] για την ταξινόμηση εικόνων εξωτικών πουλιών. Η συλλογή του εμπειρικού υλικού της έρευνας πραγματοποιήθηκε μέσω εικόνων που λήφθηκαν από διάφορες πηγές. Ο σκοπός της παρούσας πτυχιακής είναι να συγκριθούν ποιοτικά και ποσοτικά διεργασίες αναγνώρισης ειδών εξωτικών πουλιών μέσω της χρήσης τεχνολογιών τεχνητής νοημοσύνης. Ο στόχος της πτυχιακής είναι η ανάπτυξη ενός συστήματος αναγνώρισης πουλιών με ακρίβεια και αποδοτικότητα. Επιμέρους στόχοι της έρευνας περιλαμβάνουν τη βελτιστοποίηση του συστήματος που χρησιμοποιήσαμε, την επιδίωση μεγαλύτερης ακρίβειας των αποτελεσμάτων και τέλος, την αξιολόγηση της απόδοσης του συστήματος. Η καινοτομία της μελέτης έγκειται στον τρόπο με τον οποίο γίνεται η χρήση του Coral Edge TPU για την επιτάχυνση και βελτίωση της διαδικασίας αναγνώρισης ειδών πουλιών. Τα επιστημονικά ερωτήματα της εργασίας είναι:

Q1) Μπορεί το Coral Edge TPU να αναγνωρίσει σε φωτογραφίες ή εικόνες, είδη πουλιών με σχετική ακρίβεια;

Q2) Ποια είναι η ταχύτητα αναγνώρισης του Edge TPU σε σχέση με παραδοσιακούς επεξεργαστές;

Q3) Ποιο είναι το ποσοστό σφάλματος στις ταξινομήσεις;

Q4) Πώς επηρεάζει το μέγεθος της εικόνας την απόδοση του συστήματος;

Q5) Μπορεί το σύστημα να διακρίνει είδη πουλιών σε πραγματικό χρόνο;

Q6) Ποιες βελτιώσεις μπορούν να γίνουν για τη μείωση του ποσοστού σφάλματος;

Q7) Πώς συγκρίνεται η απόδοση του συστήματος με άλλες υπάρχουσες λύσεις;

Ως προς τους περιορισμούς της μελέτης διαπιστώσαμε την περιορισμένη βάση δεδομένων εικόνων και την εξάρτηση από την ακρίβεια των ετικετών. Τα ευρήματα της συγκεκριμένης μελέτης μπορούν να αξιοποιηθούν για την ανάπτυξη πιο αποδοτικών αλγορίθμων αναγνώρισης, καθώς και για την εφαρμογή των τεχνικών αναγνώρισης σε άλλους τομείς όπως η ιατρική και η βιομηχανία. Τα συμπεράσματα δείχνουν ότι η χρήση του Coral Edge TPU μπορεί να βελτιώσει την αποδοτικότητα και την ακρίβεια των συστημάτων αναγνώρισης εικόνας.

1.3 Ιστορική αναδρομή.

Η πρώτης γενιάς TPU (2016) είναι μια μηχανή πολλαπλασιασμού μήτρας 8-bit, που οδηγείται με οδηγίες CISC από τον κεντρικό επεξεργαστή σε έναν δίαυλο PCIe 3.0.Κατασκευάζεται σε διαδικασία 28 nm με μέγεθος καλουπιού ≤ 33 mm2. Η ταχύτητα ρολογιού είναι 700 MHz και έχει ισχύ θερμικής σχεδίασης 28–40W. Διαθέτει 28MiB μνήμης στο τσιπ και 4 MiB συσσωρευτών 32-bit που λαμβάνουν τα αποτελέσματα μιας συστολικής συστοιχίας 256×256 8- bit.

πολλαπλασιαστές. Στο πακέτο TPU υπάρχουν 8GiB DDR3 SDRAM διπλού καναλιού 2133MHz που προσφέρει εύρος ζώνης 34GB/s. Οι εντολές μεταφέρουν δεδομένα προς ή από τον κεντρικό υπολογιστή, εκτελούν πολλαπλασιασμούς ή συνελίξεις πινάκων και εφαρμόζουν συναρτήσεις ενεργοποίησης.2. Η δεύτερη γενιά TPU ανακοινώθηκε τον Μάιο του 2017. Ο σχεδιασμός TPU της πρώτης γενιάς περιοριζόταν από το εύρος ζώνης μνήμης και η χρήση 16 GB μνήμης υψηλού εύρους ζώνης στη σχεδίαση δεύτερης γενιάς αύξησε το εύρος ζώνης στα 600GB/s και την απόδοση στα 45 teraFLOPS. Στη συνέχεια, οι TPU διατάσσονται σε μονάδες τεσσάρων τσιπ με απόδοση 180teraFLOPS. Στη συνέχεια, 64 από αυτές τις μονάδες συναρμολογούνται σε pods 256 chip με απόδοση 11,5petaFLOPS [3].

Συγκεκριμένα, ενώ οι TPU πρώτης γενιάς περιορίζονταν σε ακέραιους αριθμούς, οι TPU δεύτερης γενιάς μπορούν επίσης να υπολογίζουν σε κινητή υποδιαστολή. Αυτό καθιστά τα TPU δεύτερης γενιάς χρήσιμα τόσο για εκπαίδευση όσο και για εξαγωγή συμπερασμάτων μοντέλων μηχανικής εκμάθησης.3. Η τρίτη γενιά επεξεργαστών TPU (2018) είναι δύο φορές πιο ισχυροί από τους TPU δεύτερης γενιάς και θα αναπτυχθούν σε pods με τέσσερις φορές περισσότερα τσιπ από την προηγούμενη γενιά. Αυτό έχει ως αποτέλεσμα 8 φορές αύξηση στην απόδοση ανά pod (με έως και 1.024 μάρκες ανά pod) σε σύγκριση με την ανάπτυξη TPU δεύτερης γενιάς.4. Η τέταρτη γενιά επεξεργαστών TPU ανακοινώθηκε τον Μάιο του 2021. Η 4η έκδοση των TPU βελτίωσε την απόδοση περισσότερο από το διπλάσιο σε σχέση με τα τσιπ της 3ηςγενιάς. Ένα μεμονωμένο v4 pod περιέχει 4.096 chips v4 και κάθε pod έχει δεκαπλάσια φορές το εύρος ζώνης διασύνδεσης ανά τσιπ σε κλίμακα, σε σύγκριση με οποιαδήποτε άλλη τεχνολογία δικτύωσης.

Το TPU αποτελείται από διάφορα κυκλώματα που συνεργάζονται για να επιταχύνουν εφαρμογές μηχανικής μάθησης. Τα κύρια κυκλώματα είναι:

•Μονάδα Matrix Multiply (MMU): Εκτελεί πράξεις πολλαπλασιασμού πινάκων, οι οποίες αποτελούν θεμελιώδεις πράξεις σε εφαρμογές μηχανικής μάθησης.

•Μονάδα Συνελίξεων (CU): Εκτελεί convolutions, οι οποίες χρησιμοποιούνται σε εφαρμογές αναγνώρισης εικόνας και επεξεργασίας σήματος.

•Μονάδα Ενεργοποίησης (AU): Εφαρμόζει μη γραμμικές συναρτήσεις ενεργοποίησης, όπως ReLU ή sigmoid, στα αποτελέσματα των MMUs και CUs.3

•Μονάδα Buffer: Αποθηκεύει δεδομένα που χρησιμοποιούνται από τις MMUs, CUs, καιAUs.

•Μονάδα Ελέγχου: Συντονίζει τη λειτουργία των υπόλοιπων κυκλωμάτων.

Η διαδικασία που ακολουθείται είναι
1.Φόρτωση δεδομένων: Τα δεδομένα εισάγονται στην TPU και αποθηκεύονται στην Μονάδα Buffer.
2.Υπολογισμοί: Η MMU εκτελεί πράξεις πολλαπλασιασμού πινάκων, ενώ η CU εκτελεί convolutions. Η AU εφαρμόζει συναρτήσεις ενεργοποίησης.
3.Αποθήκευση αποτελεσμάτων: Τα αποτελέσματα αποθηκεύονται στην ΜονάδαBuffer ή εξάγονται από την TPU.

Για την ολοκλήρωση των εργασιών επιτελείται παράλληλη επεξεργασία:
•Η TPU μπορεί να εκτελέσει πολλαπλές πράξεις ταυτόχρονα, χάρη στην ύπαρξη πολλαπλών MMUs, CUs, και AUs. Αυτό επιτρέπει την ταχεία επεξεργασία μεγάλων όγκων δεδομένων. Ως προς την ενεργειακή αποδοτικότητα γνωρίζουμε ότι η TPU σχεδιάστηκε για να είναι ενεργειακά αποδοτική.
•Χρησιμοποιεί διάφορες τεχνικές για να μειώσει την κατανάλωση ενέργειας, όπως clockgating και power gating.

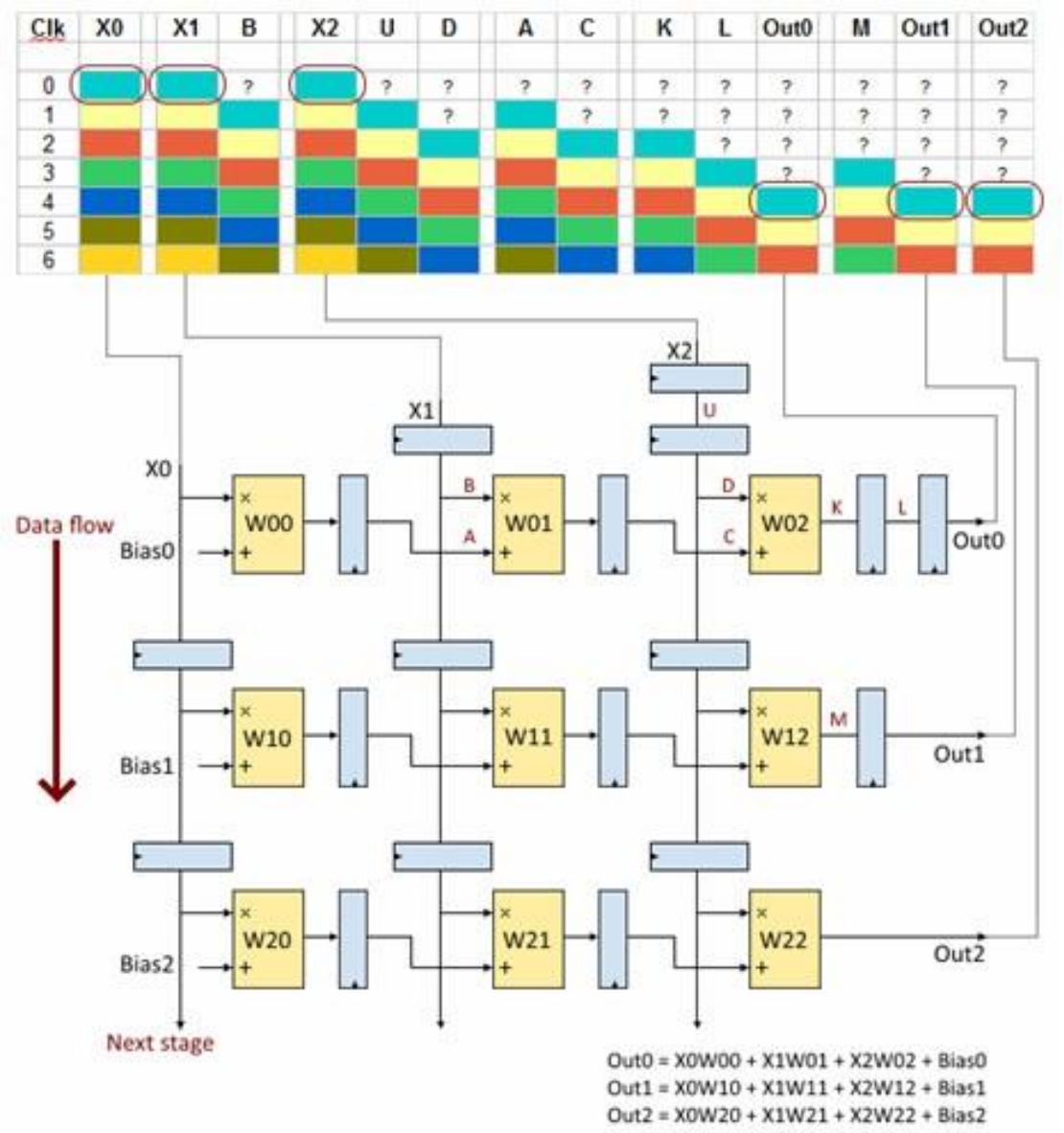
1.4 Ο USB Accelerator

Η τεχνητή νοημοσύνη έχει εισέλθει σε όλες τις πτυχές της ζωής μας, από την έξυπνη στοίβα στο σπίτι μέχρι την αυτόνομη οδήγηση. Ωστόσο, η εκτέλεση πολύπλοκων μοντέλων μηχανικής μάθησης απαιτεί συχνά τεράστιους υπολογιστικούς πόρους [4]. Εδώ είναι όπου έρχεται σε βοήθεια η συνεχής εξέλιξη των τεχνολογιών επιτάχυνσης, όπως οι USB Accelerators (Εικόνα 1). Η βιβλιοθήκη TensorFlow Lite αποτελεί μια από τις πιο δημοφιλείς επιλογές για την

ενσωμάτωση μοντέλων μηχανικής μάθησης σε φορητές συσκευές. Από τη στιγμή που η τεχνολογία αναπτύσσεται με ραγδαίους ρυθμούς, έχουν εμφανιστεί και εναλλακτικές μεθόδους για την επιτάχυνση της εκτέλεσης αυτών των μοντέλων. Ένα τέτοιο παράδειγμα είναι ο USB Accelerator της Coral, ο οποίος προσφέρει επιτάχυνση υπολογισμού για μοντέλα μηχανικής μάθησης μέσω της χρήσης ειδικού υλικού [5]. Η συνδυασμένη χρήση της βιβλιοθήκης TensorFlow Lite και του USB Accelerator της Coral ανοίγει νέους ορίζοντες για την ανάπτυξη φορητών εφαρμογών τεχνητής νοημοσύνης που απαιτούν υψηλή απόδοση. Επιτρέπει στους προγραμματιστές να εκτελούν πολύπλοκα μοντέλα μηχανικής μάθησης ακόμα και σε συσκευές με περιορισμένους υπολογιστικούς πόρους. Η εγκατάσταση και η ρύθμιση της βιβλιοθήκης TensorFlow Lite για να λειτουργεί με τον USB Accelerator της Coral είναι σχετικά απλή και αποτελεσματική. Με τη χρήση τηςTensorFlow Lite, οι προγραμματιστές μπορούν να μετατρέψουν τα μοντέλα τους σε μορφή που είναι συμβατή με το USB Accelerator και να τα ενσωματώσουν στις εφαρμογές τους με ελάχιστη προσπάθεια [6].

Εικόνα 1: USB Accelerators

Ένα από τα μεγάλα πλεονεκτήματα της χρήσης του USB Accelerator της Coral είναι η υψηλή ταχύτητα επεξεργασίας που προσφέρει. Αυτό επιτρέπει την εκτέλεση πολύπλοκων μοντέλων μηχανικής μάθησης με μικρό χρόνο απόκρισης, κάτι που είναι κρίσιμο για πολλές εφαρμογές στις οποίες απαιτείται γρήγορη ανάλυση δεδομένων. Πέρα από την ταχύτητα, ο USB Accelerator της Coral προσφέρει και υψηλή απόδοση ενέργειας. Αυτό το καθιστά ιδανικό για φορητές συσκευές και εφαρμογές που λειτουργούν με μπαταρίες, καθώς εξασφαλίζει μεγαλύτερη διάρκεια ζωής της μπαταρίας. Το συνολικό αποτέλεσμα είναι η δημιουργία πιο αποδοτικών και γρήγορων φορητών εφαρμογών τεχνητής νοημοσύνης. Από την ανίχνευση αντικειμένων έως την αναγνώριση φωνής, ο συνδυασμός της TensorFlow Lite με το USB Accelerator της Coral ανοίγει νέους ορίζοντες για την εφαρμογή της τεχνητής νοημοσύνης σε περισσότερους τομείς της καθημερινής ζωής. Συνολικά, η συνδυασμένη χρήση της βιβλιοθήκης TensorFlow Lite και του USB Accelerator της Coral αποτελεί έναν ισχυρό σύμμαχο για τους προγραμματιστές που αναζητούν γρήγορες, αποδοτικές και φορητές λύσεις τεχνητής νοημοσύνης (Εικόνα 2). Με την συνεχή εξέλιξη της τεχνολογίας, αναμένεται να δούμε ακόμα περισσότερες καινοτόμες εφαρμογές που εκμεταλλεύονται αυτήν την ισχυρή συνεργασία [7].



Εικόνα 2: Η αρχιτεκτονικη του Coral TPU

1.5 Συμπεράσματα

Στο πρώτο κεφάλαιο της πτυχιακής περιγράψαμε την γενική κατεύθυνση της πτυχιακης εργασίας, αναφέροντας τον σκοπό, τους στόχους και τα προσδοκώμενα αποτελέσματα. Σημειώσαμε την ιστορική διαδρομή του θέματος σε σχέση με την ανάπτυξη της τεχνολογίας και περιγράψαμε την λειτουργία του TPU, σε συδυασμό με τις μεθόδους ακολουθήθηκαν στην έρευνα μας. Τα κυκλώματα του TPU συνεργάζονται για να επιταχύνουν εφαρμογές μηχανικής μάθησης. Η παράλληλη επεξεργασία και η ενεργειακή αποδοτικότητα καθιστούν το TPU ένα ισχυρό εργαλείο για την ανάπτυξη και χρήση εφαρμογών τεχνητής νοημοσύνης σε συνδυασμό με τους USB Accelerators.

2. Τα χαρακτηριστικά του TensorFlow Lite και η μεταφορά μάθησης

2.1: Εισαγωγή

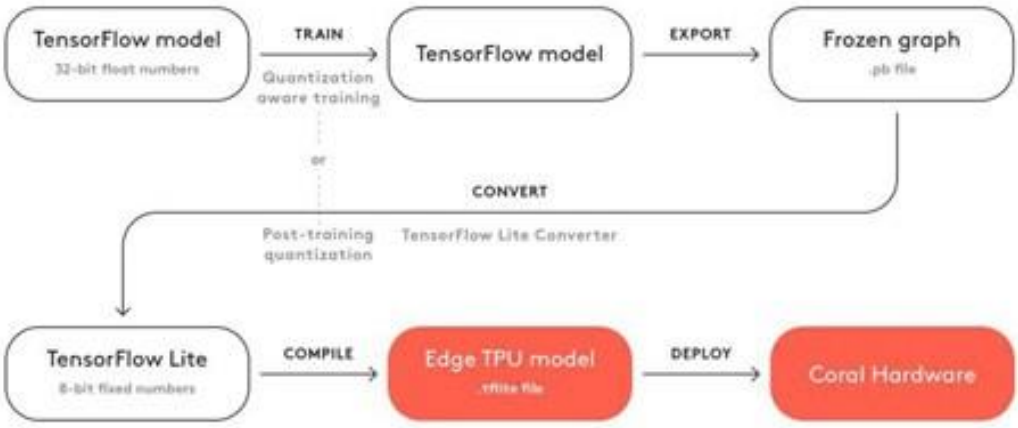
Σε αυτό το κεφάλαιο εξετάζουμε το τα τεχνικά χαρακτηριστικά του TensorFlow Lite (2.2), και παραθέτουμε μία μελέτη συμβατοτητας (2.3). Στο υποκεφάλαιο 2.4, περιγράφουμε τον τρόπο επανεκπαίδευσης ενός υπάρχοντος μοντέλου με μάθηση μεταφοράς. Τέλος, παραθέτουμε τα συμπεράσματα που προκύπτουν από το κεφάλαιο (2.5).

2.2: Τα τεχνικά χαρακτηριστικά του TensorFlow Lite

Το TensorFlow Lite είναι ένα ελαφρύ και ευέλικτο πλαίσιο ανοιχτού κώδικα που αναπτύχθηκε από την Google για την ανάπτυξη εφαρμογών μηχανικής μάθησης σε διάφορες πλατφόρμες, συμπεριλαμβανομένων των κινητών συσκευών, των ενσωματωμένων συστημάτων και των μικροελεγκτών [8]. Αυτό το πλαίσιο παρέχει απλές δομές για την ενσωμάτωση μοντέλων μηχανικής μάθησης σε εφαρμογές, ενώ παρέχει επίσης βελτιστοποιημένες λειτουργίες για την εκτέλεση αυτών των μοντέλων σε περιορισμένους πόρους περιβάλλοντα. Ένα από τα σημαντικότερα πλεονεκτήματα του TensorFlow Lite είναι η δυνατότητα να εκτελείται σε περιβάλλοντα με περιορισμένους πόρους, όπως οι κινητές συσκευές. Αυτό το καθιστά ιδανικό για ανάπτυξη εφαρμογών μηχανικής μάθησης που απαιτούν χαμηλή κατανάλωση ενέργειας και χαμηλή καθυστέρηση στην απόκριση. Επιπλέον, το TensorFlow Lite παρέχει ένα ευρύ φάσμα εργαλείων ανάπτυξης που επιτρέπουν στους προγραμματιστές να δημιουργήσουν, να εκπαιδεύσουν και να ενσωματώσουν μοντέλα μηχανικής μάθησης στις εφαρμογές τους με ευκολία. Αυτά τα εργαλεία περιλαμβάνουν το TensorFlow Lite Converter, το οποίο μετατρέπει μοντέλαTensorFlow σε μορφή που μπορεί να χρησιμοποιηθεί από το TensorFlow Lite, και το4TensorFlow Lite Interpreter, το οποίο διερμηνεύει και εκτελεί τα μοντέλα αυτά σε διάφορεςπλατφόρμες [9]. Συνολικά, το TensorFlow Lite και τα σχετικά εργαλεία ανάπτυξης παρέχουν ένα ισχυρό και ευέλικτο πλαίσιο για την ανάπτυξη εφαρμογών μηχανικής μάθησης που μπορούν να εκτελεστούν αποτελεσματικά σε περιβάλλοντα με περιορισμένους πόρους και να ενσωματωθούν σε μια ευρεία γκάμα εφαρμογών.

2.3: Επισκόπηση συμβατότητας

Η Edge TPU είναι ικανή να εκτελεί βαθιά νευρωνικά δίκτυα τροφοδότησης προς τα εμπρός, όπως τα νευρωνικά δίκτυα συνελίξεων (CNN). Υποστηρίζει μόνο τα μοντέλα TensorFlow Lite που είναι πλήρως κβαντισμένα στα 8 bit και στη συνέχεια μεταγλωττισμένα ειδικά για την Edge TPU (Εικόνα 3). Πρόκειται για μια ελαφριά έκδοση του TensorFlow που έχει σχεδιαστεί για κινητές και ενσωματωμένες συσκευές.



Εικόνα 3: Αλγόριθμος μετατροπής για τον TensorFlow Lite.

Η συγκεκριμένη έκδοση επιτυγχάνει εξαγωγή συμπερασμάτων με χαμηλή καθυστέρηση σε μικρό δυαδικό μέγεθος - τόσο τα μοντέλα Tensor Flow Lite όσο και οι πυρήνες διερμηνευτών είναι πολύ μικρότεροι. Τα μοντέλα TensorFlow Lite μπορούν να γίνουν ακόμη μικρότερα και πιο αποδοτικά μέσω κβαντισμού, ο οποίος μετατρέπει τα δεδομένα παραμέτρων 32-bit σε αναπαραστάσεις 8-bit (που απαιτούνται από την Edge TPU). Καθώς δεν μπορούμε να εκπαιδεύσουμε ένα μοντέλο απευθείας με το TensorFlow Lite- αντίθετα, πρέπει να μετατρέψετε το μοντέλο σας από ένα αρχείο TensorFlow (όπως ένα αρχείο .pb) σε ένα αρχείο TensorFlow Lite (ένα αρχείο .tflite), χρησιμοποιήσαμε τον μετατροπέα TensorFlow Lite. (Εικόνα 1). Η εικόνα 1 απεικονίζει τη βασική διαδικασία για τη δημιουργία ενός μοντέλου που είναι συμβατό με το Edge TPU. Το μεγαλύτερο μέρος της ροής εργασίας χρησιμοποιεί τα τυπικά εργαλεία TensorFlow. Για το μοντέλο TensorFlow Lite, στη συνέχεια χρησιμοποιήσαμε τον μεταγλωττιστή Edge TPU για να δημιουργήσετε ένα αρχείο .tflite που είναι συμβατό με το Edge TPU [10].

Ωστόσο, δεν χρειάστηκε να ακολουθήσουμε όλη αυτή τη διαδικασία για να δημιουργήσετε ένα καλό μοντέλο για το Edge TPU. Αξιοποιήσαμε υπάρχοντα μοντέλα TensorFlow που είναι συμβατά με την Edge TPU, επανεκπαιδεύοντάς τα με το δικό μας σύνολο δεδομένων. Για

παράδειγμα, το MobileNet είναι μια δημοφιλής αρχιτεκτονική μοντέλου ταξινόμησης/ ανίχνευσης εικόνων που είναι συμβατή με την Edge TPU. Δημιουργήσαμε διάφορες εκδόσεις αυτού του μοντέλου τις οποίοες χρησιμοποιήσαμε ως σημείο εκκίνησης για να δημιουργήσουμε το δικό μας μοντέλο αναγνώρισης. Στο επόμενο υποκεφάλαιο περιγράφουμε τον τρόπο επανεκπαίδευσης ενός υπάρχοντος μοντέλου με μάθηση μεταφοράς.

2.4: Εκμάθηση μεταφοράς στη μηχανική μάθηση

Αντί να δημιουργήσουμε το δικό σας μοντέλο και να το εκπαιδεύσουμε από την αρχή, μπορούμε να εκπαιδεύσουμε εκ νέου ένα υπάρχον μοντέλο που είναι ήδη συμβατό με την Edge TPU, χρησιμοποιώντας μια τεχνική που ονομάζεται εκμάθηση μεταφοράς (μερικές φορές ονομάζεται επίσης "λεπτή ρύθμιση"). Η εκπαίδευση ενός νευρωνικού δικτύου από το μηδέν (όταν δεν έχει υπολογισμένα βάρη ή προκατάληψη) μπορεί να διαρκέσει μέρες υπολογιστικού χρόνου και απαιτεί τεράστιο όγκο δεδομένων εκπαίδευσης [11]. Η μάθηση μεταφοράς όμως μας επιτρέπει να ξεκινήσουμε ένα μοντέλο που έχει ήδη εκπαιδευτεί για μια σχετική εργασία και στη συνέχεια να εκτελέσετε περαιτέρω εκπαίδευση για να διδάξετε στο μοντέλο νέες ταξινομήσεις χρησιμοποιώντας ένα μικρότερο σύνολο δεδομένων εκπαίδευσης. Μπορούμε να το κάνουμε αυτό επανεκπαιδεύοντας ολόκληρο το μοντέλο (προσαρμόζοντας τα βάρη σε ολόκληρο το δίκτυο), αλλά μπορούμε επίσης να επιτύχουμε πολύ ακριβή αποτελέσματα αφαιρώντας απλώς το τελικό στρώμα που εκτελεί την ταξινόμηση και εκπαιδεύοντας από πάνω ένα νέο στρώμα που αναγνωρίζει τις νέες σας κλάσεις. Χρησιμοποιώντας αυτή τη διαδικασία, με επαρκή δεδομένα εκπαίδευσης και κάποιες προσαρμογές στις υπερπαραμέτρους, μπορούμε να δημιουργήσουμε ένα πολύ ακριβές μοντέλο TensorFlow σε μία μόνο συνεδρίαση. Μόλις μείνουμε ευχαριστημένοι με την απόδοση του μοντέλου, απλά μετατρέψτε το σε TensorFlow Lite και στη συνέχεια μεταγλωττίζουμε την Edge TPU [12]. Και επειδή η αρχιτεκτονική του μοντέλου δεν αλλάζει κατά τη διάρκεια της εκμάθησης μεταφοράς, γνωρίζουμε ότι θα μεταγλωττιστεί πλήρως για την Edge TPU, με το δεδομένο ότι ξεκινάμε με ένα συμβατό μοντέλο. Για να ξεκινήσουμε χωρίς καμία ρύθμιση, δοκιμάσαμε σενάρια επανεκπαίδευσης του Google Colab. Όλα αυτά τα σενάρια εκτελούν εκμάθηση μεταφοράς σε σημειωματάρια Jupyter που φιλοξενούνται στο cloud.

2.5: Συμπεράσματα

Σε αυτό το κεφάλαιο εξετάζουμε το τα τεχνικά χαρακτηριστικά του TensorFlow Lite (2.2), και παραθέσαμε μία μελέτη συμβατοτητας (2.3). Στο υποκεφάλαιο 2.4, περιγράψαμε τρόπο επανεκπαίδευσης ενός υπάρχοντος μοντέλου με μάθηση μεταφοράς.

3. Coral Tensor Processing Units

3.1: Εισαγωγή

Σε αυτό το κεφάλαιο εξετάζουμε την αρχιτεκτονική των Coral Tensor Processing Units (3.2), και παραθέσαμε στοιχεία για τις διαδικασίες των μεταγλωτίσσεων (3.3). Στο υποκεφάλαιο 3.4, αναφέρουμε τα πλεονεκτήματα από την χρήση της μεθόδου. Τα συμπεράσματα βρίσκονται στο κεφάλαιο (3.5).

3.2: Η αρχιτεκτονική των Coral Tensor Processing Units

Η αρχιτεκτονική των Coral TPUs (Tensor Processing Units) αποτελεί ένα σημαντικό κομμάτι της υποδομής που προσφέρει υψηλή απόδοση στον τομέα της επεξεργασίας μηχανικής μάθησης [13]. Οι Coral TPUs σχεδιάστηκαν από την Google ως μέρος της προσπάθειάς της να προωθήσει την υιοθέτηση της τεχνητής νοημοσύνης σε μικρές και φορητές συσκευές. Αν και η Google δεν έχει δημοσιοποιήσει αναλυτικές πληροφορίες για την ακριβή αρχιτεκτονική των Coral TPUs, γνωρίζουμε μερικά βασικά χαρακτηριστικά και λειτουργίες τους: Υψηλή Απόδοση: Οι Coral TPUs σχεδιάστηκαν για υψηλή απόδοση στην εκτέλεση μοντέλων μηχανικής μάθησης. Η αρχιτεκτονική τους επιτρέπει την επιτάχυνση των υπολογισμών τους, επιτρέποντας έτσι την αποτελεσματική εκτέλεση μεγάλων μοντέλων. Χαμηλή κατανάλωση: Παρά την υψηλή τους απόδοση, οι Coral TPUs έχουν χαμηλή κατανάλωση ενέργειας. Αυτό τους καθιστά ιδανικούς για εφαρμογές που λειτουργούν με μπαταρίες ή έχουν περιορισμένη πρόσβαση σε ενέργεια.Υποστήριξη TensorFlow: Οι Coral TPUs συνεργάζονται στενά με το TensorFlow, ένα από τα πιο δημοφιλή πλαίσια μηχανικής μάθησης. Αυτό επιτρέπει στους προγραμματιστές να εκμεταλλευτούν τις δυνατότητες των Coral TPUs με ευκολία, ενσωματώνοτας τις στις εφαρμογές

τους με ελάχιστη προσπάθεια. Ευελιξία: Οι Coral TPUs είναι σχεδιασμένοι να προσφέρουν ευελιξία στους προγραμματιστές. Μπορούν να χρησιμοποιηθούν σε μια ευρεία γκάμα εφαρμογών, από έξυπνες κάμερες έως φορητές συσκευές αυτόνομης οδήγησης. Συνολικά, η αρχιτεκτονική των Coral TPUs συνδυάζει υψηλή απόδοση, χαμηλή κατανάλωση ενέργειας και ευελιξία, καθιστώντας τους ιδανικούς για φορητές εφαρμογές μηχανικής μάθησης σε περιβάλλοντα με περιορισμένους υπολογιστικούς πόρους.

3.3: Μεταγλώττιση

Αφού εκπαιδεύσαμε και μετατρέψαμε το μοντέλο μας σε TensorFlow Lite (με κβαντισμό), το τελικό βήμα είναι η μεταγλώττισή του με τον μεταγλωττιστή Edge TPU (Εικόνα 4).

Εικόνα 4: Μεταγλώτιση

Σε περίπτωση που το μοντέλο δεν πληροί όλες τις απαιτήσεις μπορεί να μεταγλωττιστεί, αλλά μόνο ένα μέρος του μοντέλου θα εκτελεστεί στην Edge TPU [14]. Στο πρώτο σημείο στο γράφημα του μοντέλου όπου εμφανίζεται μια μη υποστηριζόμενη λειτουργία, ο μεταγλωττιστής χωρίζει το γράφημα σε δύο μέρη. Το πρώτο μέρος του γραφήματος περιέχει μόνο υποστηριζόμενες λειτουργίες μεταγλωττίζεται σε μια προσαρμοσμένη λειτουργία που εκτελείται στην Edge TPU και όλα τα υπόλοιπα εκτελούνται [15].

3.4: Πλεονεκτήματα

Εκπαιδευτικά οφέλη:

•Βελτιωμένη απόδοση: Τα USB accelerators επιταχύνουν δραματικά την απόδοση των υπολογιστών, επιτρέποντας στους μαθητές ομαλή και παραγωγική εμπειρία μάθησης, χωρίς περισπασμούς από τεχνικά προβλήματα.

•Ευκαιρίες για STEM: Τα USB accelerators μπορούν να αξιοποιηθούν για πλήθος εκπαιδευτικών δραστηριοτήτων STEM, όπως προγραμματισμός, ρομποτική, 3Dεκτύπωση, επεξεργασία εικόνων και βίντεο, και πολλά άλλα.

•Πρόσβαση σε σύγχρονες τεχνολογίες: Η χρήση USB accelerators εξοικειώνει τους μαθητές με σύγχρονες τεχνολογίες και τους προετοιμάζει για το μέλλον της εργασίας, όπου η τεχνητή νοημοσύνη και η μηχανική μάθηση διαδραματίζουν ολοένα και σημαντικότερο ρόλο.Πρακτικά οφέλη:

•Χαμηλό κόστος: Τα USB accelerators είναι σημαντικά οικονομικά σε σχέση με τις παραδοσιακές κάρτες γραφικών, προσφέροντας ισοδύναμη ή και ανώτερη απόδοση.

•Ευκολία χρήσης: Η εγκατάσταση και η χρήση των USB accelerators είναι εύκολη και δεν απαιτούν ειδικές γνώσεις.

•Φορητότητα: Τα USB accelerators είναι πολύ φορητά και μπορούν να μεταφέρονται εύκολα από υπολογιστή σε υπολογιστή.

•Ευελιξία: Τα USB accelerators μπορούν να χρησιμοποιηθούν σε οποιοδήποτε USB port,σε οποιοδήποτε λειτουργικό σύστημα (Windows, macOS, Linux, ChromeOS).

•Μείωση κατανάλωσης ενέργειας: Τα USB accelerators καταναλώνουν λιγότερη ενέργεια σε σχέση με τις παραδοσιακές κάρτες γραφικών, συμβάλλοντας στην εξοικονόμηση ενέργειας.

•Αθόρυβη λειτουργία: Τα USB accelerators λειτουργούν αθόρυβα και δεν παράγουν θερμότητα, συμβάλλοντας σε ένα πιο ήσυχο και άνετο περιβάλλον μάθησης.

3.5: Συμπεράσματα

Σε αυτό το κεφάλαιο εξετάσαμε την αρχιτεκτονική των Coral Tensor Processing Units (3.2), και παραθέσαμε στοιχεία για τις διαδικασίες των μεταγλωτίσσεων (3.3). Στο υποκεφάλαιο 3.4, αναφέραμε τα πλεονεκτήματα από την χρήση της μεθόδου.

4. Λογισμικά, εγκαταστάσεις, συγκρίσεις και μετρήσεις

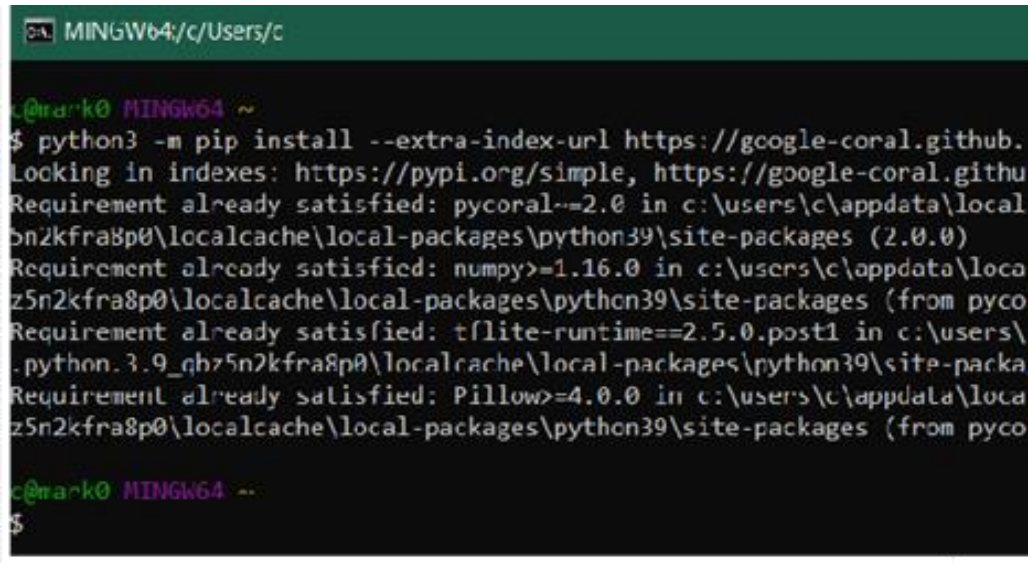
4.1: Εισαγωγή

Σε αυτό το κεφάλαιο εξετάζουμε την αρχιτεκτονική του Coral TPU (4.2), και παραθέσαμε στοιχεία πραδείγματα και Bench Markings σε περιβάλλοντα Windows για τις διαδικασίες των μεταγλωτίσσεων (4.3). Στο υποκεφάλαιο 4.4, περιγράφουμε την αξιοποίηση του Raspberry PI και αναφερόμαστε στις τελικές συγκρίσεις, από την χρήση της μεθόδου. Τα συμπεράσματα βρίσκονται στο κεφάλαιο (4.5).

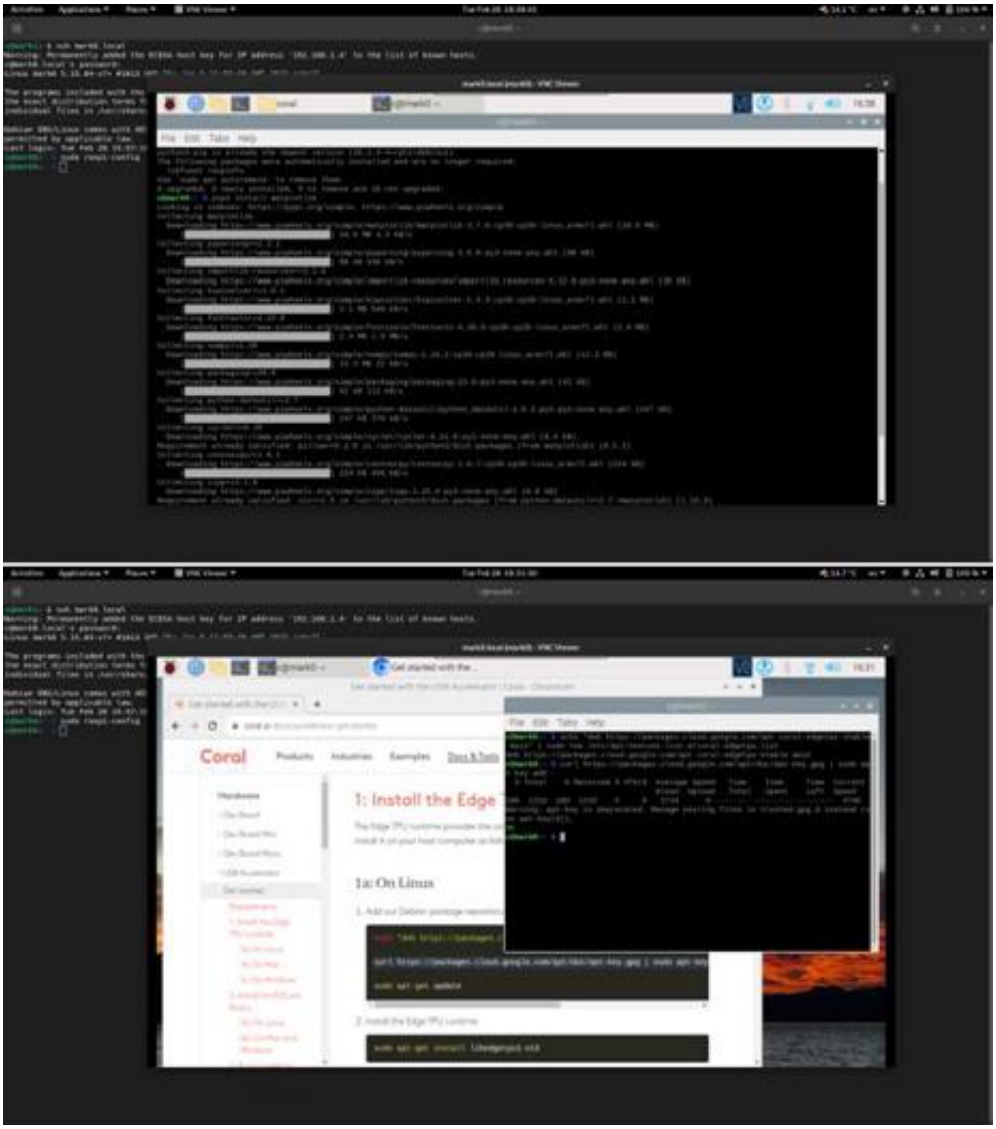
4.2: Η διαδικασία εγκατάστασης των προτεινόμενων λογισμικών

Η διαδικασία εγκατάστασης είναι εξαρτάται από την έκδοση του λειτουργικού στα Linux. Το Edge TPU υποστηρίζει εκδόσεις της python μεταξύ των 3.6 και 3.9. Οτιδήποτε νεότερο ή παλιότερο δεν θα αφήσει τα μοντέλα να «τρέξουν» σ τ η ν π λ α κ έ τ α . Μ π ο ρ ε ί ν α χρησιμοποιηθεί εικονικό περιβάλλον για στηθεί μία από τις συμβατές εκδόσεις στην περίπτωση που το λειτουργικό έχει προγκατεστημένη νεότερη έκδοση της python. Η διαδικασία είναι περίπλοκη και μακροσκελής, αλλά υπάρχει εκτενής οδηγός: pyenv.Windows 10, αλλά υποστηρίζονται και Windows 11. Η εγκατάσταση γίνεται με βάση Python εκδόσεις που υποστηρίζονται 3.6 – 3.9. Η εγκατάσταση ενός Bash κάνει την διαδικασία της εκτέλεσης των μοντέλων και την εγκατάσταση διάφορων βιβλιοθηκών αρκετά εύκολη. Για την παρούσα μελέτη χρησιμοποιήσαμε το git για Windows (64 bit έκδοση). Για αυτήν την χρήση απαιτείται η βιβλιοθήκη της C++. Από github κατεβάσαμε και τρέξαμε τον φάκελο (Εικόνα 5) και εκτελέσαμε το install.batPyCoral. Ανοίγοντας το Git, «τρέξαμε» την εντολή “python3 -m pip install --extra-index-url https://google-coral.github.io/py-repo/”, (Εικόνα 6) και πήραμε το αποτελέσματα της Εικόνας 7.

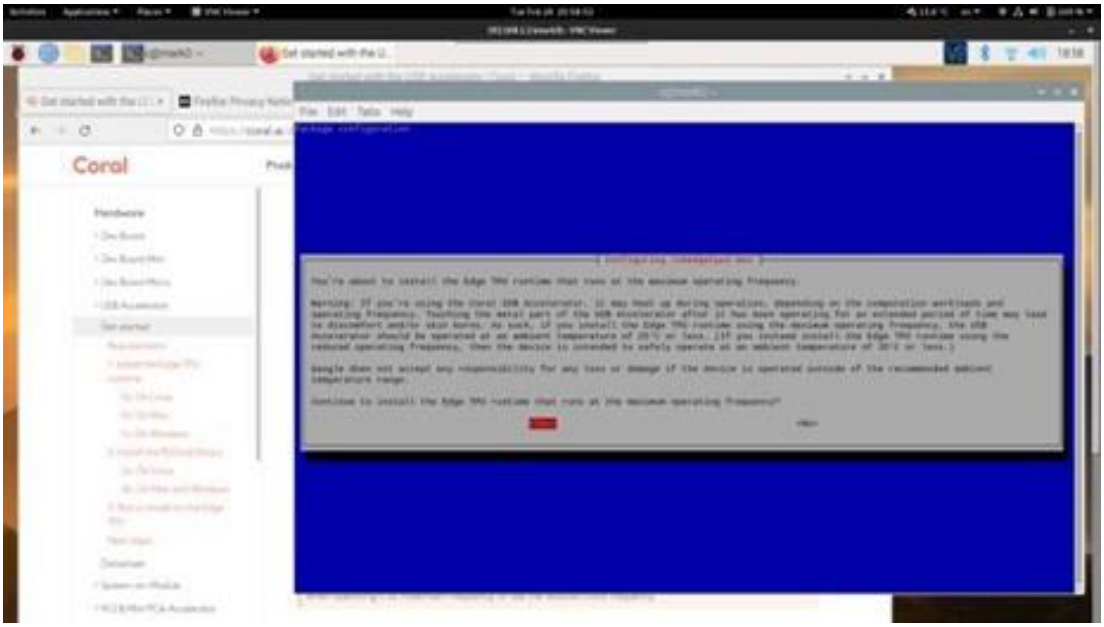
Εικόνα 5: Διαδικασίες εγκατάστασης



Εικόνα 6: Εντολοδότηση



Εικόνα 7: Αποτελέσματα των διαδικασιών εγκατάστασης.

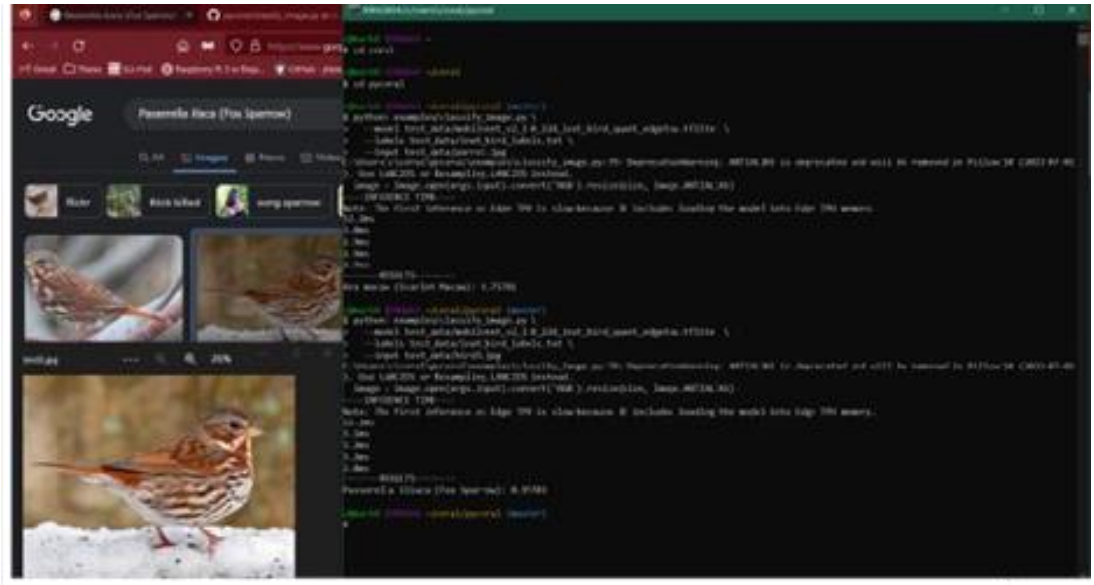


Εικόνα 8: Διαδικασίες εγκατάστασης

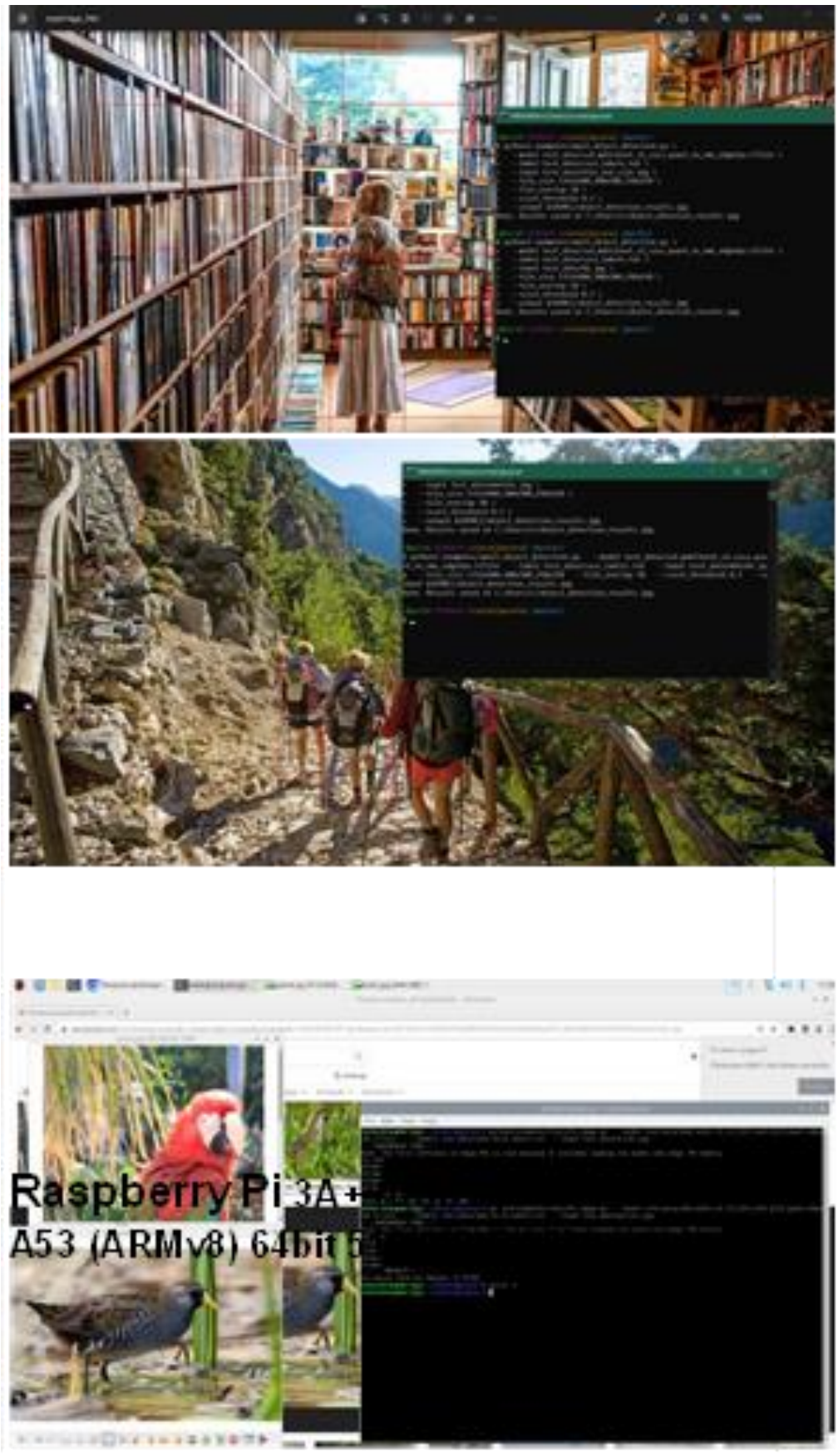
Κατά την εγκατάσταση δίνεται η δυνατότητα ο TPU να τρέχει με τον μέγιστο αριθμό στο ρολόι του (sudo apt-get install libedgetpu1-max). Για επαναφορά (sudo apt-get install) (Εικόνα 8).

4.3: Παραδείγματα και Bench Markings Windows

Οι οθόνες εργασίας για τις τελικές σταθμίσεις δίνονται στην Εικόνα 9 και 10.



Εικόνα 9: Κώδικας και οθόνη εργασίας



Εικόνα 10: Οθόνες εργασίας

4.4: Αξιοποίηση του Raspberry PI

Επεκτείνοντας την μελέτη οργανώσαμε μία πειραματική διάταξη για την μέτρηση της απόδοσης 4 μοντέλων (Pi 4 4GB - 8GB , Pi 3B και Pi 3A+) του Raspberry PI. Η απόδοση μετριέται με και χωρίς επιταχυντή Coral USB. Το ίδιο σύνολο σεναρίων Python χρησιμοποιείται για την εκτέλεση ταξινόμησης εικόνων χρησιμοποιώντας ένα μοντέλο μηχανικής εκμάθησης (MobileNet V1) σε όλα τα μοντέλα. Αυτό επιτυγχάνεται με την εναλλαγή της ίδιας κάρτας micro SD μεταξύ των διαφορετικών παραλλαγών.

Η ρύθμιση (Εικόνα 11) αφορά:

- clasiffy.py χρησιμοποιεί το mobilenet_v1_1.0_224_quant.tflite μοντέλο και δεν απαιτεί το coral usb
- clasiffy_coral.py χρησιμοποιεί το mobilenet_v1_1.0_224_quant_edgetpu.tflite μοντέλο και κάνει χρήση του usb accelerator

Η διαδικασία που εκτελούν και τα 2 προγράμματα είναι πανομοιότυπη

1. Εισαγάγαμε το load_delgate από το tflite_runtime.interpreter
2. Αλλάξαμε τη διαδρομή του αρχείου μοντέλου και κατεύθυνση προς το να δείχνει στο αρχείο μοντέλου 'edgetpu'.
3. Δημιουργήσαμε διερμηνέα με συνάρτηση 'load_delegate'.

Χωρίς TPU	Με TPU
Raspberry Pi 4B (4GB)	

>>> 56.5 ms (camera capture)
>>> 16.21 ms (inference)
>>> 130.26 ms (preview)
tennis ball 0.9411764705882353
	>>> 68.61 ms (camera capture)
	>>> 155.29 ms (inference)
	>>> 133.59 ms (preview)
	tennis ball 0.8666666666666667

Raspberry Pi 4B (8GB)	

>>> 67.21 ms (camera capture)
>>> 7.7 ms (inference)
>>> 141.56 ms (preview)
tennis ball 0.8784313725490196
	>>> 69.88 ms (camera capture)
	>>> 150.67 ms (inference)
	>>> 131.96 ms (preview)
	tennis ball 0.8313725490196079

Raspberry Pi 3B	

>>> 85.49 ms (camera capture)
>>> 358.56 ms (inference)
>>> 208.69 ms (preview)
tennis ball 0.9764705882352941
	>>> 80.6 ms (camera capture)
	>>> 19.5 ms (inference)
	>>> 354.38 ms (preview)
	tennis ball 0.8274509803921568

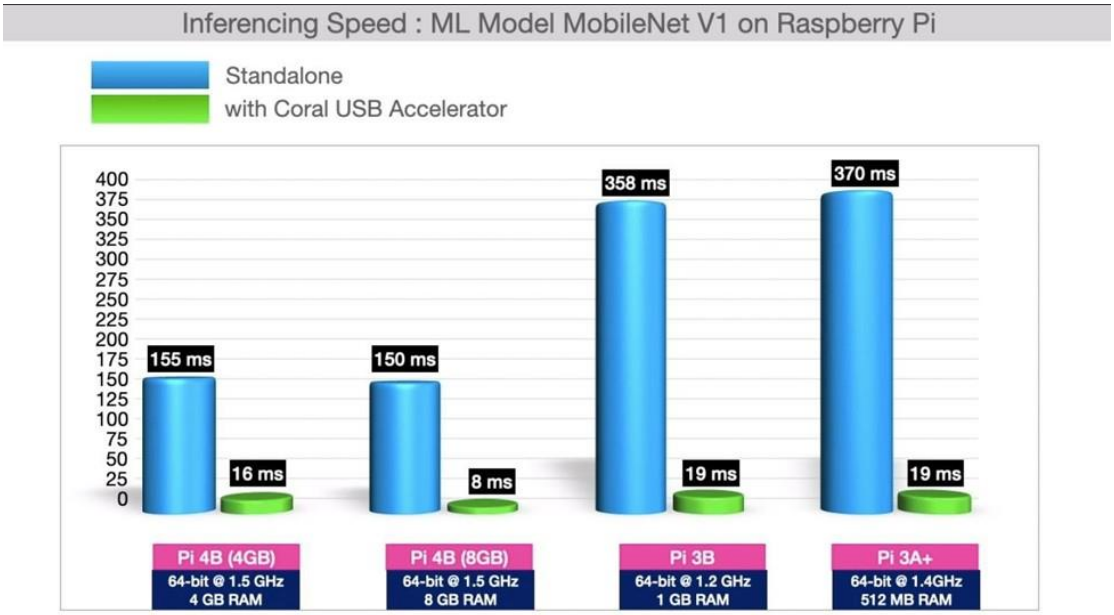
Raspberry Pi 3A+	

>>> 79.81 ms (camera capture)
>>> 19.32 ms (inference)
>>> 205.34 ms (preview)
tennis ball 0.9137254901960784
	>>> 84.35 ms (camera capture)
	>>> 374.26 ms (inference)
	>>> 221.63 ms (preview)
	tennis ball 0.8549019607843137

Εικόνα 11: Οθόνες εργασίας για συγκρίσει λειτουργιών

Η “Λήψη κάμερας” και η “Προεπισκόπηση” περιλαμβάνουν τη λήψη μιας εικόνας(frame) από την κάμερα και την εμφάνισή της σε ένα παράθυρο εξόδου. Υπάρχουν πολλές μέθοδοι. Μια τέτοια μέθοδος είναι να εκτελέσετε την εργασία που σχετίζεται με την κάμερα μέσω της βιβλιοθήκης OpenCV. Το συμπέρασμα περιλαμβάνει τη λήψη προβλέψεων από το αρχείο μοντέλου με βάση την εικόνα εισόδου. Ο χρόνος που απαιτείται σε αυτό το βήμα εξαρτάται από το αρχείο μοντέλου που χρησιμοποιείται. Ο χρόνος συμπερασμάτων μπορεί να διαφέρει από μοντέλο σε μοντέλο ανάλογα με το πόσες κλάσεις έχει. Χωρίς εξωτερική επιτάχυνση υλικού, αυτή η εργασία εκτελείται από την CPU και καταναλώνει σχεδόν όλους τους πόρους του επεξεργαστή.

Προκειμένου να δημιουργηθούν εφαρμογές που χρησιμοποιούν ένα μοντέλο μηχανικής εκμάθησης για μια περίπτωση χρήσης σε πραγματικό χρόνο, είναι επιτακτική ανάγκη ο χρόνος εξαγωγής συμπερασμάτων να είναι όσο το δυνατόν χαμηλότερος για να επιτευχθεί μέγιστο FPS. Το πρόγραμμα Python classify_coral.py “τοποθετεί” το τμήμα συμπερασμάτων στον επιταχυντή USB Coral και μειώνει δραστικά τον χρόνο επεξεργασίας. Σε όλες τις παραπάνω περιπτώσεις, παρατηρούμε τη δραστική μείωση του χρόνου συμπερασμάτων κατά την χρήση του Coral. Ωστόσο, η “λήψη κάμερας” και η “προεπισκόπηση” εξακολουθούν να απαιτούν τον ίδιο χρόνο, επειδή μόνο το τμήμα εξαγωγής συμπερασμάτων επεξεργάζεται μέσα στο υλικό Coral. Μια επισκόπηση των αποτελεσμάτων παρέχεται το γράφημα της Εικόνας 12.



Εικόνα 12: Γράφημα αποτύπωσης απόδοσης λειτουργίας

Ο κώδικας υλοποιεί ένα παράδειγμα που εκτελεί ανίχνευση αντικειμένων στα πλαίσια κάμερας χρησιμοποιώντας τη βιβλιοθήκη OpenCV. Πριν την εκτέλεση του κώδικα, χρειάζεται να δηλωθούν τα εξής:

- TEST_DATA: η διαδρομή του φακέλου που περιέχει τα μοντέλα ανίχνευσης αντικειμένων και τις ετικέτες.
- Περιεχόμενα TEST_DATA: το μοντέλο ανίχνευσης προσώπου "mobilenet_ssd_v2_face_quant_postprocess_edge tpu.tflite" και το μοντέλο ανίχνευσης αντικειμένων "mobilenet_ssd_v2_coco_quant_postprocess_edge tpu.tflite", καθώς και το αρχείο ετικετών "coco_labels.txt" που περιέχει τις ετικέτες για το μοντέλο αντικειμένων.

Ο κώδικας (Εικόνα 13) χρησιμοποιεί τον πίνακα argparse για την είσοδο παραμέτρων κατά το ξεκίνημα του προγράμματος, έτσι ώστε να μπορεί να διαλέξει το μοντέλο ανίχνευσης που θα χρησιμοποιηθεί και να καθορίσει το όριο σκορ ανίχνευσης που χρησιμοποιείται για την απόφαση αντικειμένου. Στη συνέχεια, ο κώδικας ανοίγει την κάμερα, διαβάζει συνεχώς τα καρέ της κάμερας, επεξεργάζεται κάθε καρέ με το μοντέλο ανίχνευσης και εμφανίζει τα αποτελέσματα στην οθόνη. Η ανίχνευση αντικειμένων γίνεται σε πραγματικό χρόνο. Αυτή η συνάρτηση παίρνει μια εικόνα σε μορφή OpenCV (cv2_im), μέγεθος εισόδου (inference_size), αντικείμενα (objs) και ετικέτες (labels) ως είσοδο και προσθέτει πλαίσια γύρω από τα αντικείμενα και ετικέτες πάνω τους. Αντί για την αρχική ανάλυση της εικόνας, οι ανιχνευτές αντικειμένων συνήθως λειτουργούν σε μια μειωμένη ανάλυση (inference_size) για να βελτιώσουν την ταχύτητα της ανίχνευσης. Επομένως, η συνάρτηση αυτή υπολογίζει το κλιμάκωση x και y των αντικειμένων στην αρχική ανάλυση της εικόνας και στη συνέχεια τα αντιστοιχεί στο αρχικό μέγεθος της εικόνας. Στη συνέχεια, για κάθε αντικείμενο στη λίστα των αντικειμένων (objs), υπολογίζει τις συντεταγμένες του πλαισίου (bbox), προσθέτει το πλαίσιο στην εικόνα (cv2.rectangle), και προσθέτει την ετικέτα του αντικειμένου πάνω στο πλαίσιο (cv2.putText). Τέλος, επιστρέφει την εικόνα με τα πλαίσια και τις ετικέτες.

```python
import argparse
import cv2
import os

from pycoral.adapters.common import input_size
from pycoral.adapters.detect import get_objects
from pycoral.utils.dataset import read_label_file
from pycoral.utils.edgetpu import make_interpreter
from pycoral.utils.edgetpu import run_inference


def main():
    #Define os path
    default_model_dir = '../all_models'
    #trained models
    default_model = 'mobilenet_ssd_v2_coco_quant_postprocess_edgetpu.tflite'
    #labels
    default_labels = 'coco_labels.txt'

    parser = argparse.ArgumentParser()
    parser.add_argument('--model', help='.tflite model path',
                        default=os.path.join(default_model_dir,default_model))
    parser.add_argument('--labels', help='label file path',
                        default=os.path.join(default_model_dir, default_labels))
    parser.add_argument('--top_k', type=int, default=3,
                        help='number of categories with highest score to display')
    parser.add_argument('--camera_idx', type=int, help='Index of which video source to use. ', default = 0)
    parser.add_argument('--threshold', type=float, default=0.1,
                        help='classifier score threshold')
    args = parser.parse_args()

    print('Loading {} with {} labels.'.format(args.model, args.labels))
    interpreter = make_interpreter(args.model)
    interpreter.allocate_tensors()
    labels = read_label_file(args.labels)
    inference_size = input_size(interpreter)
    #enable camera driver
    cap = cv2.VideoCapture(args.camera_idx)

    #Check if camera is enabled, then create windows with live output
    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break
        cv2_im = frame

        cv2_im_rgb = cv2.cvtColor(cv2_im, cv2.COLOR_BGR2RGB)
        cv2_im_rgb = cv2.resize(cv2_im_rgb, inference_size)
        run_inference(interpreter, cv2_im_rgb.tobytes())
        objs = get_objects(interpreter, args.threshold)[:args.top_k]
        cv2_im = append_objs_to_img(cv2_im, inference_size, objs, labels)

        cv2.imshow('frame', cv2_im)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    cv2.destroyAllWindows()

def append_objs_to_img(cv2_im, inference_size, objs, labels):
    height, width, channels = cv2_im.shape
    scale_x, scale_y = width / inference_size[0], height / inference_size[1]
    for obj in objs:
        bbox = obj.bbox.scale(scale_x, scale_y)

        #define statistics
        x0, y0 = int(bbox.xmin), int(bbox.ymin)
        x1, y1 = int(bbox.xmax), int(bbox.ymax)
        #print recognition score
        percent = int(100 * obj.score)
        #import object from txt
        label = '{}% {}'.format(percent, labels.get(obj.id, obj.id))

        #Define object recognition shape and colors
        cv2_im = cv2.rectangle(cv2_im, (x0, y0), (x1, y1), (0, 0, 0), 2)
        #Text from labels.txt output on cv2 image
        cv2_im = cv2.putText(cv2_im, label, (x0, y0+30),
                             cv2.FONT_HERSHEY_SIMPLEX, 1.0, (255, 0, 0), 2)
    return cv2_im

if __name__ == '__main__':
    main()
```

Εικόνα 13: Κώδικας λειτουργίας

4.5: Διδακτική Προσέγγιση

Η διδασκαλία της αναγνώρισης αντικειμένων και ζώων μέσω του **Google Coral TPU** στα Επαγγελματικά Λύκεια (ΕΠΑ.Λ.) μπορεί να πραγματοποιηθεί σε διακριτά στάδια,

ενσωματώνοντας τόσο θεωρητική κατάρτιση όσο και πρακτική εφαρμογή. Αρχικά, οι μαθητές θα εισαχθούν στις βασικές αρχές της τεχνητής νοημοσύνης και των τεχνητών νευρωνικών δικτύων, με έμφαση στη λειτουργία και τα οφέλη της **Edge Computing** και της **εκμάθησης μεταφοράς** (transfer learning). Οι μαθητές θα διδαχθούν πώς τα προεκπαιδευμένα μοντέλα μηχανικής μάθησης μπορούν να προσαρμοστούν σε νέες εφαρμογές, μειώνοντας τον απαιτούμενο χρόνο και πόρους εκπαίδευσης ενός δικτύου από το μηδέν .
Μετά τη θεωρητική εισαγωγή, θα ακολουθήσει η πρακτική εφαρμογή. Οι μαθητές θα χωριστούν σε ομάδες και θα εργαστούν πάνω σε ένα προκαθορισμένο σενάριο αναγνώρισης αντικειμένων ή ζώων. Χρησιμοποιώντας το **Google Coral TPU**, οι ομάδες θα συνδέσουν το hardware με το λογισμικό, θα εισαγάγουν έτοιμα δεδομένα εικόνων και θα εκπαιδεύσουν το μοντέλο τους να αναγνωρίζει νέες κλάσεις αντικειμένων ή ζώων. Αυτή η διαδικασία περιλαμβάνει την αφαίρεση του τελικού επιπέδου ταξινόμησης από ένα προεκπαιδευμένο νευρωνικό δίκτυο και την προσαρμογή του στις νέες ανάγκες μέσω εκπαίδευσης μεταφοράς, χρησιμοποιώντας ένα μικρότερο σύνολο δεδομένων .
Κατά τη διάρκεια της υλοποίησης, οι μαθητές θα καθοδηγούνται στη ρύθμιση των υπερπαραμέτρων του μοντέλου και στη διαδικασία επανεκπαίδευσης, προκειμένου να βελτιώσουν την ακρίβεια αναγνώρισης. Κάθε ομάδα θα παρουσιάσει τα αποτελέσματά της στην τάξη, προκειμένου να αξιολογηθεί τόσο η ορθότητα του προγραμματισμού όσο και η κατανόηση των θεωρητικών εννοιών. Η διδακτική αυτή μέθοδος επιτρέπει στους μαθητές να συνδέσουν την πρακτική με τη θεωρία, ενισχύοντας την κατανόηση τους στις νέες τεχνολογίες αναγνώρισης, και τους παρέχει τις απαραίτητες δεξιότητες για τον προγραμματισμό εξατομικευμένων ηλεκτρονικών συσκευών .

4.6: Συμπεράσματα

Σε αυτό το κεφάλαιο εξετάσαμε την αρχιτεκτονική του Coral TPU (4.2), και παραθέσαμε στοιχεία πραδείγματα και Bench Markings σε περιβάλλοντα Windows για τις διαδικασίες των μεταγλωτίσσεων (4.3). Στο υποκεφάλαιο 4.4, περιγράψαμε την αξιοποίηση του Raspberry PI και αναφερθήκαμε στις τελικές συγκρίσεις, από την χρήση της μεθόδου.

Με την ολοκλήρωση της διδακτικής ενότητας, οι μαθητές θα έχουν αναπτύξει τόσο θεωρητικές γνώσεις όσο και πρακτικές δεξιότητες στον τομέα της αναγνώρισης αντικειμένων και ζώων μέσω του **Google Coral TPU**. Θα κατανοούν τις βασικές αρχές της τεχνητής νοημοσύνης και των τεχνητών νευρωνικών δικτύων, καθώς και την έννοια της **εκμάθησης μεταφοράς** και τη σημασία της στην επανεκπαίδευση προεκπαιδευμένων μοντέλων για νέες εφαρμογές. Πρακτικά, θα είναι σε θέση να ρυθμίζουν και να συνδέουν το hardware με το λογισμικό, να εισάγουν δεδομένα για την εκπαίδευση νευρωνικών δικτύων και να προσαρμόζουν μοντέλα αναγνώρισης σε διαφορετικές ταξινομήσεις. Επιπλέον, οι μαθητές θα αναπτύξουν δεξιότητες στην ομαδική εργασία και στην επίλυση προβλημάτων μέσω της συνεργασίας σε ομάδες, ενώ η παρουσίαση των αποτελεσμάτων τους θα ενισχύσει την ικανότητά τους να επικοινωνούν τεχνικές έννοιες με σαφήνεια. Τέλος, οι μαθητές θα έχουν αποκτήσει τις απαραίτητες βάσεις για να εφαρμόσουν τις γνώσεις τους σε πραγματικά προβλήματα, κάτι που θα τους βοηθήσει στην επαγγελματική τους εξέλιξη στον τομέα της πληροφορικής και των εξατομικευμένων ηλεκτρονικών συσκευών .

Βιβλιογραφία – Δικτυογραφία

[1] https://ai.googleblog.com/2019/11/introd ucing-next-generation-on-device.Html
[2] https://en.wikipedia.org/wiki/Tensor_Pro cessing_Unit
[3] https://en.wikipedia.org/wiki/Application-specific_integrated_circuit [4]https:// en.wikipedia.org/wiki/AI_accelera tor
[5] https://www.techradar.com/news/computi ng-components/processors/google-s-tensorprocessing-unit-explained-this-is- what-the-future-of-computing-looks-like-1326915
[6] https://cloud.google.com/tpu/docs/system -architecture-tpu-vm?hl=en
[7] https://coral.ai/docs/accelerator/datas heet/https://dspace.lib.ntua.gr/xmlui/hand le/ 123456789/55937
[8] https://coral.ai/docs/m2/datasheet/htt ps://el.wikipedia.org/wiki/Tensor_Flow

[9] https://google.github.io/flatbuffers/
[10] https://arxiv.org/pdf/2102.10423.pdf [XI]https://towardsdatascience.com/a-basic-introduction-to-tensorflow-lite- 59e480c57292
[11] https://github.com/jiteshsaini/coral_ USB_ml_accelerator
[12] https://helloworld.co.in/article/image
-classification-tensorflow-lite
[13] https://github.com/EdjeElectronics/T ensorFlow-Lite-Object-Detection-on-Android-and-Raspberry-Pi
[14] https://coral.ai/docs/edgetpu/models- intro/#transfer-learning
[15] https://coral.ai/docs/edgetpu/models- intro/#compatibility-overview