



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών &
Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Διπλωματική Εργασία

Need a title for the Thesis!

Εκπόνηση:
Παναγιώτου Κωνσταντίνος
ΑΕΜ: 7316

Επίβλεψη:
Αν. Καθ. Πέτρου Λουκάς
Υπ. Δρ. Μουσουλιώτης
Παναγιώτης

Θεσσαλονίκη, Σεπτέμβριος 2016

The robots are going to help us find our crystal...

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστίες here!

Περίληψη

Title

Thesis Title here!

Abstract

Konstantinos Panayiotou
Electrical & Computer Engineering Department,
Aristotle University of Thessaloniki, Greece
September 2016

Περιεχόμενα

Ευχαριστίες	iii
Περίληψη	v
Abstract	vii
Ακρωνύμια	x
1 Εισαγωγή	1
1.1 Περιγραφή του Προβλήματος	2
1.2 Σκοπός - Συνεισφορά της Διπλωματικής Εργασίας	2
1.3 Διάρθρωση της Αναφοράς	2
2 Επισκόπηση της Ερευνητικής Περιοχής	4
3 Τεχνικές Βαθιάς Μηχανικής Μάθησης και Αναγνώριση Αντικειμένων στον χώρο	6
3.1 Deep Learning	9
3.2 Νευρωνικά Δίκτυα Συνέλιξης	9
3.3 Αναγνώριση αντικειμένων με χρήση Νευρωνικών Δικτύων Συνέλιξης .	9
4 Εργαλεία Hardware/Software που χρησιμοποιήθηκαν	10
4.1 NVIDIA Jetson TK1 development board	10
4.2 Full software stack for developing Deep Neural Networks	13
5 Implementations	14
5.1 YOLO implementation with keras DNN framework	14
5.2 Desktop PC configuration for optimal performance	14
5.3 Jetson TK1 dev board setup and applied optimizations	14
6 Πειράματα - Αποτελέσματα	16
7 Συμπεράσματα	18
8 Μελλοντικές επεκτάσεις	20
Βιβλιογραφία	21

Κατάλογος Σχημάτων

3.1	Τεχνητή Νοημοσύνη	7
3.2	Παράδειγμα διαφορετικών αναπαραστάσεων των δεδομένων	8
4.1	Tegra K1 SOC	11
4.2	Jetson TK1 development board	12

Ακρωνύμια Εγγράφου

Παρακάτω παρατίθενται ορισμένα από τα πιο συχνά χρησιμοποιούμενα ακρωνύμια της παρούσας διπλωματικής εργασίας:

AI	→ Artificial Intelligence
ML	→ Machine Learning
NN	→ Neural Network
DL	→ Deep Learning
DNN	→ Deep Neural Network
MLP	→ Multilayer Perceptron
CNN	→ Convolutional Neural Network
ConvNet	→ Convolutional Networks
RNN	→ Recurrent Neural Network
YOLO	→ You Only Look Once
SOC	→ System On Chip
OS	→ Operating System

1

Εισαγωγή

Ο όρος *Τεχνητή Νοημοσύνη*, γενικότερα αναφέρεται στην ικανότητα των υπολογιστικών συστημάτων να μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς. Η επιθυμία των ανθρώπων να κατασκευάσουν "έξυπνες" μηχανές, καταγράφεται από τα έτη της αρχαίας Ελλάδας. Μυθικές μορφές όπως οι Πυγμαλίωνας, Δαίδαλος και Ήφαιστος μπορούν να θεωρηθούν ως θρυλικοί εφευρέτες και δημιουργοί νοούντων μηχανών όπως η Γαλάτεια, ο Τάλος και η Πανδώρα.

Ένας πιο ολοκληρωμένος ορισμός της Τεχνητή Νοημοσύνη θα ήταν:

"Ο κλάδος/τομέας της επιστήμης της πληροφορικής, που ασχολείται με την σχεδίαση και κατασκευή ευφυών συστημάτων, δηλαδή συστημάτων που κατέχουν χαρακτηριστικά που σχετίζονται με την ανθρώπινη νοημοσύνη και συμπεριφορά." Με την εμφάνιση των πρώτων ηλεκτρονικών και (επανα)-προγραμματιζόμενων υπολογιστικών συστημάτων, οι άνθρωποι ξεκίνησαν να σκέφτονται τρόπους για να κατασκευάσουν "έξυπνες" μηχανές. Η ραγδαία εξέλιξη στον κλάδο της επιστήμης της πληροφορικής, τις τελευταίες δεκαετίες, έφερε και την εξέλιξη στην επιστήμη της Τεχνητής Νοημοσύνης. Το 1997, η IBM κατασκεύασε ένα υπολογιστικό σύστημα το οποίο μπορούσε να πέξει σκάκι (Deep Blue) [1], το οποίο κέρδισε τον παγκόσμιο τότε πρωταθλητή στο σκάκι, Garry Kasparov. Το σκάκι έχει εξηντα-τέσσερις θέσεις και τριάντα-δύο πιόνια που μπορούν να κινούνται με συγκεκριμένο τρόπο. Η μηχανή Deep Blue, μπορούσε να εκτιμήσει και να αξιολογήσει διακόσια εκατομύρια πιθανές καταστάσεις της σκακιέρας. Ωστόσο, πρέπει να σημειώσουμε ότι η επίλυση του πρόβληματος του σκακιού, είναι ένα απλό πρόβλημα το οποίο μπορεί να περιγραφεί πλήρως μέσα από μια λίστα με κανόνες.

Η δυνατότητα ενός "ευφυούς" ρομποτικού συστήματος, να αντιλαμβάνεται το περιβάλλον του, είναι απαραίτητη ικανότητα που πρέπει να διαθέτει. Ένας ρομποτικός πράκτορας, σε πληθώρα εφαρμογών, πρέπει να διαθέτει τόσο την ικανότητα να αναγνωρίζει, μέσω εικόνων λήψης από κάμερας, διάφορα αντικείμενα, όσο και την ικανότητα να εντοπίζει την ακριβές θέση του εκάστοτε αντικειμένου στον χώρο. Δεν θα είχε νόημα για ένα ρομπότ, να έχει άποψη για την παρουσία αντικειμένων

γύρω του, αν δεν έχει και την ικανότητα να γνωρίζει και την θέση των αντικειμένων αυτών.

1.1 ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Ο μεγάλος χρόνος που χρειάζεται ένα CNN, τόσο για να κατηγοριοποιήσει, όσο και για να εντοπίσει, ένα αντικείμενο το κάνει ακατάλληλο για χρήση σε εφαρμογές πραγματικού χρόνου. Αυτό συμβαίνει λόγω του "βάθους" που έχουν τα νευρωνικά αυτά δίκτυα. Είναι ιδιαίτερα σημαντικό, ένα ρομπότ να μπορεί να αντιλαμβάνεται το περιβάλλον του; να μπορεί να αναγνωρίζει ανθρώπους, ζώα, αντικείμενα γενικότερα. Ωστόσο, θέλουμε τα ρομπότ να είναι και όσο πιο "ελκιστικά" γίνεται στον άνθρωπο ή/και μικρότερα. Αυτό, φέρει σαν αποτέλεσμα να μην μπορούμε να τοποθετήσουμε ογκώδη, άρα με μεγάλη υπολογιστική ισχύ, υπολογιστικά συστήματα κατευθείαν πάνω στα ρομπότ.

1.2 ΣΚΟΠΟΣ - ΣΥΝΕΙΣΦΟΡΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Η παρούσα διπλωματική έχει στόχο την υλοποίηση και σύγκριση διαφόρων μοντέλων CNN για αναγνώριση και εντοπισμό αντικειμένων σε εικόνες (object detection), σε ένα ρομποτικό σύστημα που φέρει ενσωματωμένο σύστημα.

Παρόλο που σήμερα έχουν σχεδιαστεί CNN μοντέλα που έχουν την δυνατότητα να αναγνωρίσουν και να κατηγοριοποιήσουν αντικείμενα από χιλιάδες, αν όχι και περισσότερες, κλάσεις, ο χρόνος που απαιτείται για να κατηγοριοποιήσει αντικείμενα σε μία εικόνα είναι αρκετά μεγάλος, της τάξης των δευτερολέπτων σε σύγχρονους υπολογιστές. Αυτό κάνει την χρήση CNN σε εφαρμογές πραγματικού χρόνου ακατάλληλη, όπως για παράδειγμα σε ένα ρομπότ.

1.3 ΔΙΑΦΩΡΩΣΗ ΤΗΣ ΑΝΑΦΟΡΑΣ

DOCUMENT SECTION here!

Η διάρθρωση της παρούσας διπλωματικής εργασίας είναι η εξής:

- Στο **Μέρος I** περιγράφονται ορισμένες εισαγωγικές έννοιες και βασικές γνώσεις, ώστε ο αναγνώστης να κατανοήσει με μεγαλύτερη ευκολία τα αντικείμενα που πραγματεύεται η διπλωματική εργασία. Ειδικότερα, τα κεφάλαια είναι:
 - **Κεφάλαιο 2:** Παρατίθεται η ανασκόπηση της ερευνητικής περιοχής αναφορικά με τα αντικείμενα στα οποία επιδιώκει να παρουσιάσει λύσεις η διπλωματική εργασία.
 - **Κεφάλαιο 3:** Περιγράφονται τα βασικά θεωρητικά στοιχεία στα οποία βασίστηκε η παρούσα υλοποίηση, καθώς επίσης και οι διάφορες τεχνικές και τα εργαλεία που χρησιμοποιήθηκαν.
- Το **Μέρος II** αποτελείται από τρία κεφάλαια στα οποία περιγράφονται πλήρως η υλοποιήσεις.

- Στο **Μέρος III** παρουσιάζεται αναλυτικά η μεθοδολογία των πειραμάτων, τα αποτελέσματά τους και τα συμπεράσματα στα οποία καταλήξαμε. Τέλος, γίνεται αναφορά σε διαδικαστικά προβλήματα που ανέκυψαν και προτείνονται θέματα για μελλοντική μελέτη.

2

Επισκόπηση της Ερευνητικής Περιοχής

Τόσο η αναγνώριση αντικειμένων (object recognition) όσο και η εντοπισμός της θέσης αντικειμένων, σε σχέση με (detection/localization) το αντίστοιχο σύστημα συντεταγμένων μίας εικόνας, είναι μία ερευνητική περιοχή με τεράστιο ενδιαφέρον και η οποία απασχολεί πληθώρα ερευνητών. Η επιστήμη της Μηχανικής Όρασης (Machine Learning), στοχεύει στο να δώσει λύσεις στα συγκεκριμένα προβλήματα, εισάγοντας μαθηματικά μοντέλα, τόσο αναλυτικά, όσο και πιθανοτικά.

Ο κλάδος της Βαθιάς Μηχανικής Εκμάθησης [2], ανάγει το πρόβλημα της έυρεσης χαρακτηριστικών σημείων, για την αναγνώριση αντικειμένων, στην εκμάθηση ανα-παραστάσεων [3], με την χρήση Νευρωνικών Δικτύων Συνέλιξης (CNN). Οι πρώτες εφαρμογές Νευρωνικών Δικτύων Συνέλιξης, για την αναγνώριση αντικειμένων σε εικόνες, αναπτύχθηκαν το 1990 από τον Yann LeCun. Η πιο γνωστή και επιτυχής είναι το δίκτυο LeNet [4], το οποίο χρησιμοποιήθηκε για την αναγνώριση ψηφίων σε εικόνες. Ωστόσο, η εισαγωγή των CNN στον κλάδο της Μηχανικής Όρασης έγινε το 2012 με την ανάπτυξη του δικτύου AlexNet [5], από τους Alex Krizhevsky, Ilya Sutskever και Geoffrey E. Hinton. Το δίκτυο AlexNet χρησιμοποιήθηκε στον διαγωνισμό ImageNet ILSVRC challenge, το 2012, κερδίζοντας με διαφορά 10,9%, στο σφάλμα αναγνώρισης αντικειμένων σε σύνολο 1000 κλάσεων. Με την εμφάνιση του δικτύου AlexNet, η ερευνητική κοινότητα ξεκίνησε να πιστεύει στην αποτελεσματικότητα των Νευρωνικών Δικτύων Συνέλιξης σε εφαρμογές αναγνώρισης αντικειμένων σε εικόνες. Συνέχεια στο έργο του Alex Krizhevsky έδωσαν οι ερευνητές αναπτύσσοντας, το 2013, το ZF-Net [6], το οποίο είναι βασισμένο στην αρχιτεκτονική του δικτύου AlexNet. Μέχρι σήμερα, έχουν σχεδιαστεί και αναπτυχθεί διάφορα μοντέλα Νευρωνικών Δικτύων Συνέλιξης για αναγνώριση αντικειμένων, με πιο πρόσφατο το ResNet [7], το οποίο αναπτύχθηκε από τον Kaiming He. Το ResNet (Residual Network) έχει την ιδιαιτερότητα απουσίας πλήρες συνδεδεμένων επιπέδων και είναι από τα πιο δημοφιλείς μοντέλα που εφαρμόζονται σε πρακτικά προβλήματα αναγνώρισης αντικειμένων σε εικόνες [8].

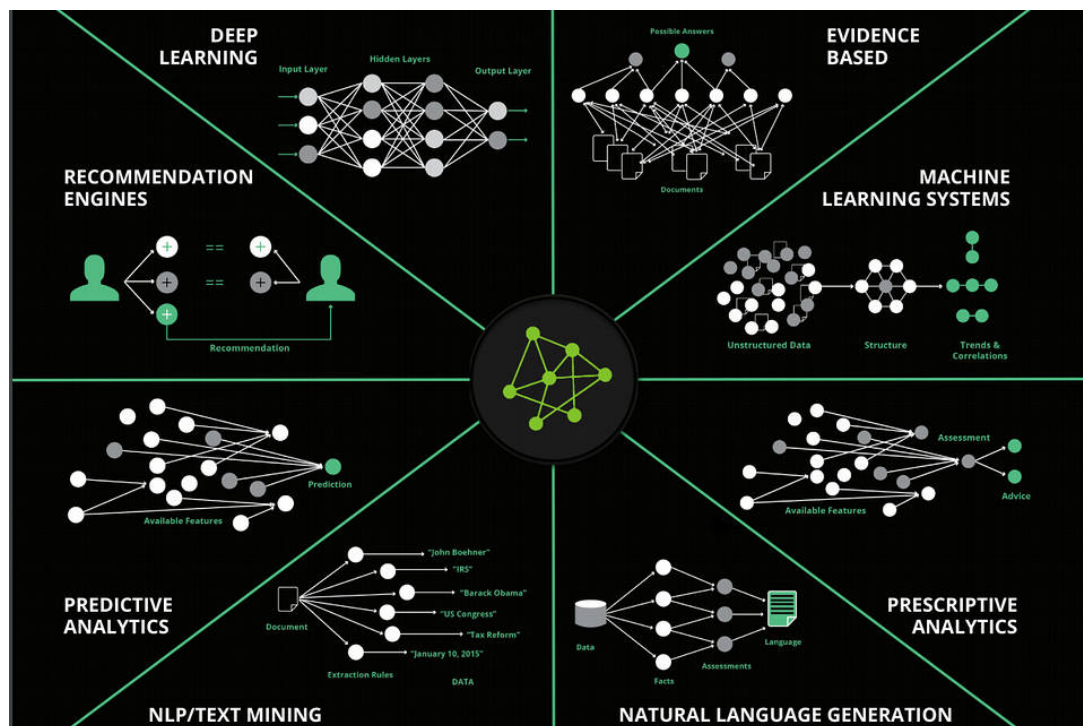
Τα προαναφερθέντα μοντέλα Νευρωνικών Δικτύων Συνέλιξης δίνουν λύσεις μόνο

στο πρόβλημα της αναγνώρισης και όχι στο πρόβλημα του εντοπισμού της θέσης των αντικειμένων αυτών. Το 2013, ερευνητές εργαζόμενοι στην Google Inc., σχεδίασαν και υλοποίησαν ένα μοντέλο Νευρωνικού Δικτύου το οποίο δίνει λύσεις στο πρόβλημα της ταυτόχρονης αναγνώρισης και εντοπισμού αντικειμένων πάνω σε πλαίσια εικόνων [9]. Το μοντέλο αυτό, που φέρει το όνομα DetectorNet, είναι ομαδική εργασία των Christian Szegedy, Alexander Toshev και Dumitru Erhan. Το μοντέλο αυτό είναι πιθανοτηκό αφού "προβλέπει" τις οριοθετημένες θέσεις για διάφορες κλάσεις αντικειμένων στον πλαίσιο μίας εικόνας. Ωστόσο, ένα βασικό μειονέκτημα του DetectorNet που το κάνει ακατάλληλο για εφαρμογή σε προβλήματα πραγματικού χρόνου, όπως για παράδειγμα σε ένα ρομποτικό σύστημα, είναι οι τεράστιες απαιτήσεις του σε πόρους και χρόνο.

3

Τεχνικές Βαθιάς Μηχανικής Μάθησης και Αναγνώριση Αντικειμένων στον χώρο

Τα τελευταία χρόνια, ο κλάδος της Τεχνητής Νοημοσύνης είναι ένας από τους πιο ραγδαία αναπτυσσόμενους κλάδους της επιστήμης της πληροφορικής με τεράστιο ερευνητικό και πρακτικό ενδιαφέρον; διαιρείται σε δύο υπο___; την συμβολική τεχνική νοημοσύνη και την υποσυμβολική τεχνητή νοημοσύνη. Η πρώτη προσπαθεί να επιλύσει τα προβλήματα χρησιμοποιώντας αλγοριθμικές διαδικασίες, δηλαδή σύμβολα και λογικούς κανόνες, ενώ η δεύτερη προσπαθεί να αναπαράγει την ανθρώπινη "ευφυΐα" μέσα από την χρήση αριθμητικών μοντέλων που με την σύνθεσή τους προσομοιώνουν την λειτουργία του ανθρώπινου εγκεφάλου (υπολογιστική νοημοσύνη). Η ικανότητα ενός νοούμενου (AI) συστήματος, να αποκτά από μόνο του γνώση, εξάγοντας πρότυπα ή/και χαρακτηριστικά σημεία από τα δεδομένα, είναι γνωστή ως *Μηχανική Μάθηση* (ML). Η εισαγωγή του κλάδου του ML στην επιστήμη των υπολογιστών, επέτρεψε στους υπολογιστές να μπορούν να αντιμετωπίσουν προβλήματα που εμπλέκουν την αντίληψη για τον πραγματικό κόσμο και να πέρνουν υποκειμενικές αποφάσεις.



Σχήμα 3.1: Τεχνητή Νοημοσύνη

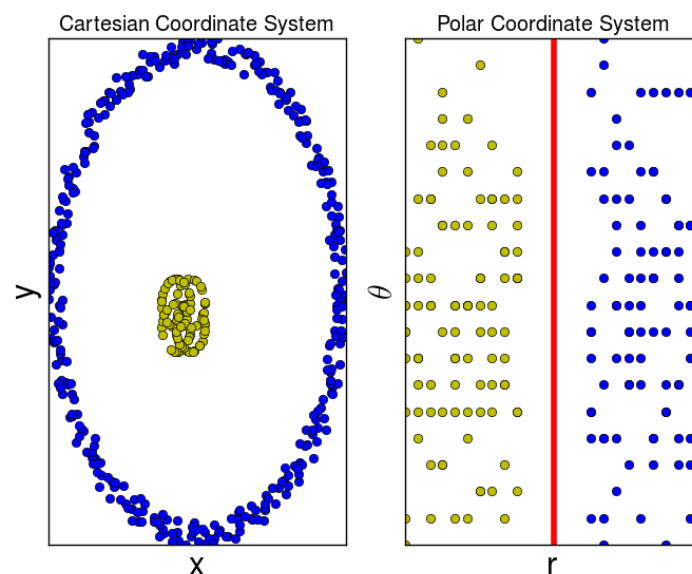
Η χρησιμοποίηση αλγορίθμων ML, επιτρέπουν σε συστήματα AI να προσαρμόζονται εύκολα σε καινούργια έργα, με ελάχιστη επέμβαση από τον άνθρωπο. Για παράδειγμα, ένα Νευρωνικό Δίκτυο που έχει εκπαιδευτεί να αναγνωρίζει γάτους σε εικόνες, δεν απαιτεί να σχεδιαστεί και να εκπαιδευτεί από το μηδέν για να έχει την ικανότητα να αναγνωρίζει και σκύλους.

Πολλά προβλήματα, ενώ μέχρι και πριν μία μερικά χρόνια λύνονταν με "χειρόγραφη", προγραμματισμένη από τον άνθρωπο, γνώση (hand-written feature extraction), σήμερα χρησιμοποιούνται αλγόριθμοι ML για την επίλυσή τους. Πιο κάτω παρατίθενται μερικά παραδείγματα:

- Αναγνώριση ομιλίας - Speech Recognition
- Μηχανική όραση - Computer Vision
 - Αναγνώριση αντικειμένων σε εικόνες - Object Recognition
 - Αναγνώριση και εντοπισμός της θέσης αντικειμένων σε εικόνες - Object Detection
- Αναγνώριση ηλεκτρονικών επιθέσεων στο διαδίκτυο - Cyberattack detection
- Επεξεργασία φυσικής γλώσσας - Natural Language Processing
 - Κατανόηση της φυσικής γλώσσας του ανθρώπου - Natural Language Understanding
 - Μοντελοποίηση και χρησιμοποίηση της φυσικής γλώσσας του ανθρώπου από μηχανές - Natural Language Generation
- Μηχανές αναζήτησης

ΚΕΦΑΛΑΙΟ 3. ΤΕΧΝΙΚΕΣ ΒΑΘΙΑΣ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΣΤΟΝ ΧΩΡΟ

Η μορφή της αναπαράστασης των δεδομένων αποτελεί σημαντικό παράγοντα στην απόδοση των αλγορίθμων ML. Μία αναπαράσταση αποτελείται από χαρακτηριστικά (features). Για παράδειγμα, ένα χρήσιμο χαρακτηριστικό, στην ταυτοποίηση ομιλιτή, από δεδομένα ήχου, είναι η εκτίμηση του μεγέθους της φωνητικής έκτασης του ομιλιτή. Έτσι, πολλά προβλήματα τεχνητής νοημοσύνης, μπορούν να λυθούν με κατάλληλη σχεδίαση και επιλογή των χαρακτηριστικών, για το συγκεκριμένο πρόβλημα. Το σύνολο των χαρακτηριστικών αυτών αποτελεί την αναπαράσταση των δεδομένων, σε ένα πιο υψηλό και αφαιρετικό επίπεδο αντίληψης για τους υπολογιστές, η οποία στην συνέχεια δίνεται σαν είσοδος σε έναν απλό ML αλγόριθμο, ο οποίος έχει μάθει να αντιστοιχεί την αναπαράσταση των δεδομένων στην επιθυμητή έξοδο.



Σχήμα 3.2: Παράδειγμα διαφορετικών αναπαραστάσεων των δεδομένων

Ένα απλό και κατανοητό παράδειγμα, το οποίο δείχνει την εξάρτηση της επίδοσης ενός αλγορίθμου ML, από την μορφή της αναπαράστασης που του δίνεται, φαίνεται στο σχήμα 3.2. Έστω ότι θέλουμε να διαχωρίσουμε τα δεδομένα μας σε δύο κλάσεις, χαράζοντας μία ευθεία μεταξύ τους. Αν αναπαραστήσουμε τα δεδομένα στο Καρτεσιανό σύστημα συντεταγμένων (αριστερό διάγραμμα), τότε η επίλυση του προβλήματος είναι αδύνατη αφού δεν υπάρχει καμία ευθεία που να διαχωρίζει τις δύο κλάσεις. Ωστόσο, αν αναπαραστήσουμε τα δεδομένα στο Πολικό σύστημα συντεταγμένων (δεξί διάγραμμα), τότε το πρόβλημα λύνεται εύκολα, χαράζοντας μία κάθετη ευθεία, με $r \in [r_1, r_2], 0 \leq \theta \leq 2\pi$.

Σε πληθώρα προβλημάτων τεχνητής νοημοσύνης, η επιλογή κατάλληλων χαρακτηριστικών είναι δύσκολο και χρονοβόρο έργο. Έστω για παράδειγμα ότι θέλουμε να αναγνωρίσουμε πρόσωπα σε εικόνες. Ένα χαρακτηριστικό θα μπορούσε να είναι τα μάτια. Δυστυχώς όμως, η αναγνώριση ματιών είναι και αυτή ένα δύσκολο πρόβλημα, αφού δεν μπορεί να περιγραφεί πάντα επακριβώς έχοντας σαν δεδομένα τις τιμές των pixel της εικόνας. Η γεωμετρική, για παράδειγμα, μορφή των ματιών

σε μία εικόνα λήψης εξαρτάται από την γωνία λήψης, τον φωτισμό, κτλ.

3.1 DEEP LEARNING

3.2 ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΣΥΝΕΛΙΞΗΣ

3.3 ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΩΝ ΜΕ ΧΡΗΣΗ ΝΕΥΡΩΝΙΚΩΝ ΔΙΚΤΥΩΝ ΣΥΝΕΛΙΞΗΣ

4

Εργαλεία Hardware/Software που χρησιμοποιήθηκαν

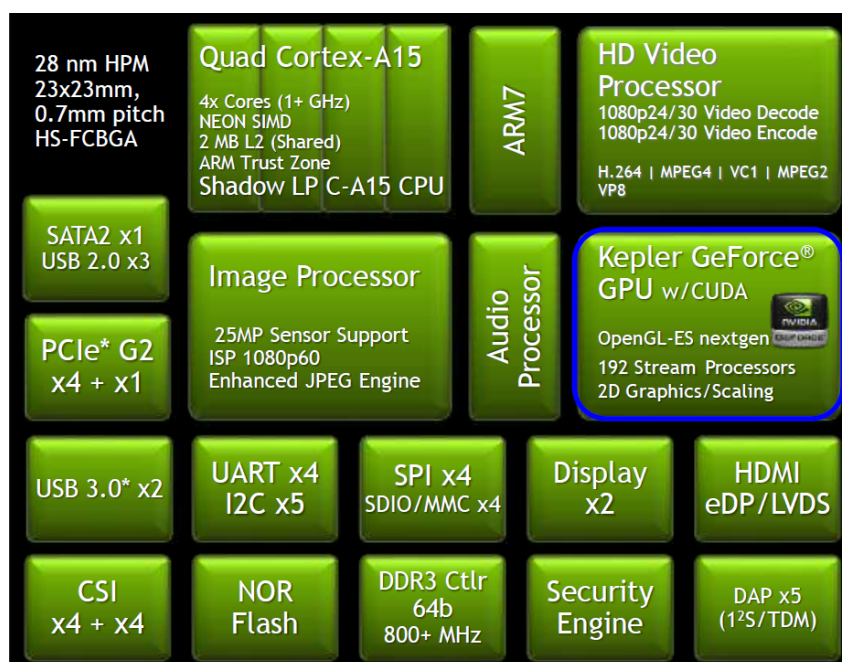
TODO Introduction!!

4.1 NVIDIA JETSON TK1 DEVELOPMENT BOARD

Στο κεφάλαιο αυτό παρουσιάζεται η ενσωματωμένη πλατφόρμα Jetson TK1 της NVIDIA, η οποία χρησιμοποιήθηκε για εφαρμογή των υλοποιήσεων για *Αναγνώριση και Εντοπισμό Αντικειμένων* σε συστήματα πραγματικού χρόνου, με Νευρωνικά Δίκτυα Συνέλιξης.

Ο *Tegra K1* είναι το πρώτο SOC της NVIDIA, για φορητές συσκευές, με προηγμένη αρχιτεκτονική και χαρακτηριστικά, καθώς και χαμηλή κατανάλωση ισχύος. Η μέγιστη κατανάλωση είναι στα *3Watt*, τιμή που δικαιολογεί την ενσωμάτωσή του σε φορητές συσκευές (tablets, smartphones), καθώς και σε προηγμένα ενσωματωμένα συστήματα (embedded systems) για εφαρμογές πραγματικού χρόνου.

4.1. NVIDIA JETSON TK1 DEVELOPMENT BOARD



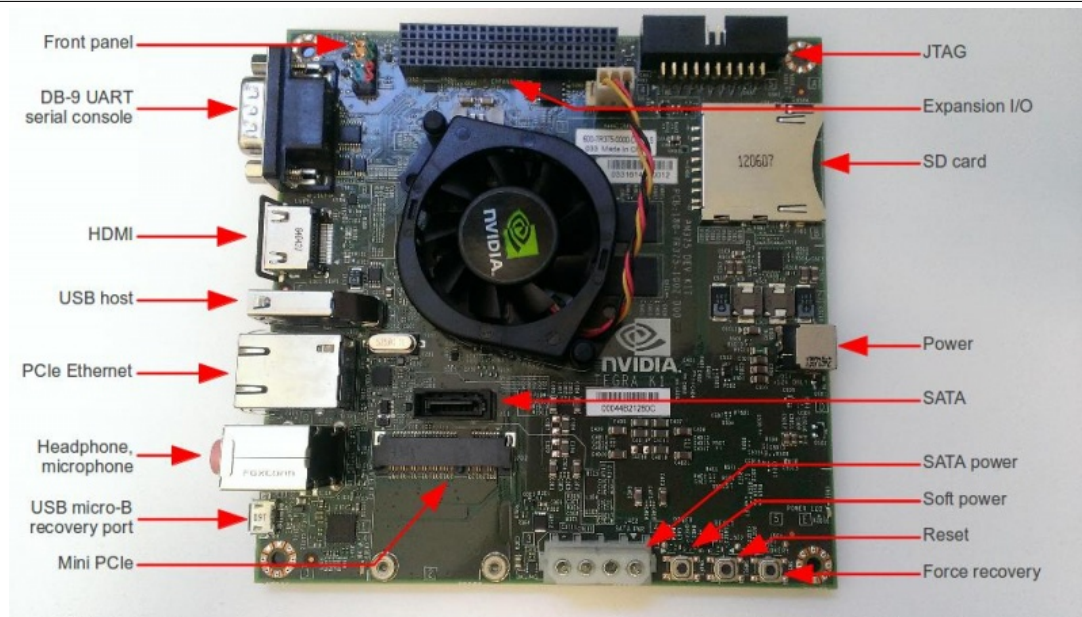
Σχήμα 4.1: Tegra K1 SOC

Όπως φαίνεται στο σχήμα 4.1, τα βασικά τεχνικά χαρακτηριστικά του Tegra K1 SOC:

- CPU: Quad-core ARM Cortex-A15 CPU, 2.3Ghz
- GPU: GK20A Kepler-based GPU with 192 CUDA cores
- RAM: DDR3L/LPDDR3, up to 8GB
- Peripherals I/O: USB, eMMC/SD-card, LVDS, HDMI, SPI, UART, I2C, SATA, PCIe
- ISP: Image processor

Στα πλαίσια της παρούσας διπλωματικής εργασίας, επιλέχτηκε να χρησιμοποιήσουμε το *Jetson TK1* development board της NVIDIA. Το Jetson TK1 ενσωματώνει το Tegra K1 SOC (CPU+GPU+ISP) και είναι πλήρως συμβατό με διάφορες διανομές λειτουργικών συστημάτων Linux (Ubuntu, Debian, Arch, Fedora, openSUSE, Gentoo). Η πλήρης συμβατότητα και υποστήριξη λειτουργικού συστήματος Linux ήταν βασικό κριτήριο στην επιλογή του συγκεκριμένου ενσωματωμένου συστήματος αφού επιτρέπει την εγκατάσταση εργαλείων με τον ίδιο τρόπο όπως σε ένα σταθερό υπολογιστικό σύστημα (Desktop PC) το οποίο τρέχει Linux OS. Πέρα από την συμβατότητα με κλασσικές διανομές Linux OS, η NVIDIA έχει αναπτύξει δικό της λειτουργικό σύστημα, *Linux4Tegra*, το οποίο έχει σαν βάση τα Ubuntu-14.04, με κάποιες επεκτάσεις για πλήρη υποστήριξη του hardware του Jetson TK1. Επιπλέον παράγοντας στην επιλογή της συγκεκριμένης πλατφόρμας είναι η πληθώρα των περιφερειακών διεπαφών που διαθέτει, κάνοντας το χρήσιμο για εφαρμογή σε ρομποτικά συστήματα όπου η σύνδεση διαφόρων περιφερειακών συσκευών, όπως

ΚΕΦΑΛΑΙΟ 4. ΕΡΓΑΛΕΙΑ HARDWARE/SOFTWARE ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ



Σχήμα 4.2: Jetson TK1 development board

για παράδειγμα αισθητήρες, κάμερες, κινητήρες, σερβο-κινητήρες, είναι απαραίτητη. Πιο κάτω δίνονται οι βασικές διεπαφές που προσφέρει η πλατφόρμα Jetson TK1:

- mini-PCIe: Σύνδεση πρόσθετων συσκευών στον δίαυλο PCI-Express όπως, Wifi cards, SSD δίσκων, κτλ.
- USB 2.0 port: Για σύνδεση συσκευών ή/και αισθητήρων με διεπαφή eHCI (Extended Host Controller Interface)
- USB 3.0 port: Για σύνδεση συσκευών ή/και αισθητήρων με διεπαφή xHCI (eXtensible Host Controller Interface)
- HDMI: Δίνει την δυνατότητα σύνδεσης οθόνης
- RS232 port: Παρόλο που το RS232 είναι αρκετά παλιό πρωτόκολλο επικοινωνίας, ακόμη ενσωματώνεται σε διάφορες συσκευές που δεν απαιτούν μεγάλο όγκο μεταφοράς δεδομένων, όπως οι οδηγοί κινητήρων
- Audio IO
- Gigabit Ethernet LAN: Δικτύωση της πλατφόρμας με τον "έξω" κόσμο
- SATA: Επιτρέπει την σύνδεση σκληρού δίσκου SATA
- JTAG port: Το JTAG προσφέρει την δυνατότητα σύνδεσης συσκευής/προγράμματος εντοπισμού σφαλμάτων (debugger), για επαγγελματική αποσφαλμάτωση
- UART port
- I2C ports: Διαθέτει τρεις θύρες I2C για σύνδεση αισθητήρων/συσκευών που οδηγούνται με το συγκεκριμένο πρωτόκολλο

4.2. FULL SOFTWARE STACK FOR DEVELOPING DEEP NEURAL NETWORKS

- GPIO: Προσφέρει δύο θύρες επέκτασης (expansion ports), με 50 και 75 ακροδέκτες αντίστοιχα. Χρησιμο κυρίως για, επικοινωνία συσκευών με SPI, οδήγηση σερβο-κινητήρων, διακλάδωση τροφοδοσίας σε τρίτες συσκευές, κτλ.

Όσον αφορά την υλοποίηση και ανάπτυξη Νευρωνικών Δικτύων σε ενσωματωμένα συστήματα, το Jetson TK1 θεωρείτε ιδανικό αφού υποστηρίζει CUDA και cuDNN. Η cuDNN (CUDA Deep Neural Network library) είναι GPU-accelerated βιβλιοθήκη για Νευρωνικά Δίκτυα, η οποία αναπτύχθηκε από την NVIDIA και προσφέρει υψηλού επιπέδου υλοποιήσεις για ρουτίνες όπως συνέλιξη, κανονικοποίηση δεδομένων, pooling, επίπεδα ενεργοποίησης, κτλ. Η βιβλιοθήκη cuDNN χρησιμοποιείτε και ενσωματώνετε σε όλα τα frameworks βαθιάς εκμάθησης, όπως Caffe [10], Tensorflow [11], Theano [12][13][14], Torch [15][16][17], CNTK και Keras.

4.2 FULL SOFTWARE STACK FOR DEVELOPING DEEP NEURAL NETWORKS

5

Implementations

TODO Introduction here!
Implementations on i7 and jetsonTK1 board

5.1 YOLO IMPLEMENTATION WITH KERAS DNN FRAMEWORK

TODO!!

5.2 DESKTOP PC CONFIGURATION FOR OPTIMAL PERFORMANCE

TODO!!

5.3 JETSON TK1 DEV BOARD SETUP AND APPLIED OPTIMIZATIONS

6

Πειράματα - Αποτελέσματα

TODO!!

7

Συμπεράσματα

TODO!!

8

Μελλοντικές επεκτάσεις

TODO!!

Βιβλιογραφία

- [1] Murray Campbell, A Joseph Hoane, and Feng-hsiung Hsu. “*Deep blue*”. Artificial intelligence, 134(1):57–83, 2002.
- [2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. “*Deep Learning*”. Book in preparation for MIT Press, 2016. URL <http://www.deeplearningbook.org>.
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. “*Representation learning: A review and new perspectives*”. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.
- [4] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “*Gradient-based learning applied to document recognition*”. Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “*ImageNet Classification with Deep Convolutional Neural Networks*”. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, “*Advances in Neural Information Processing Systems 25*”, pages 1097–1105. Curran Associates, Inc., 2012.
- [6] Matthew D. Zeiler and Rob Fergus. “*Visualizing and Understanding Convolutional Networks*”. CoRR, abs/1311.2901, 2013.
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “*Deep Residual Learning for Image Recognition*”. CoRR, abs/1512.03385, 2015.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “*Identity Mappings in Deep Residual Networks*”. CoRR, abs/1603.05027, 2016.
- [9] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. “*Deep neural networks for object detection*”. In “*Advances in Neural Information Processing Systems*”, pages 2553–2561, 2013.
- [10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. “*Caffe: Convolutional Architecture for Fast Feature Embedding*”. arXiv preprint arXiv:1408.5093, 2014.
- [11] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin

- Wicke, Yuan Yu, and Xiaoqiang Zhang. “*TensorFlow: A system for large-scale machine learning*“. CoRR, abs/1605.08695, 2016.
- [12] Rami Al-Rfou, Guillaume Alain, Amjad Almahairi, Christof Angermueller, Dzmitry Bahdanau, Nicolas Ballas, Frédéric Bastien, Justin Bayer, Anatoly Belikov, Alexander Belopolsky, Yoshua Bengio, Arnaud Bergeron, James Bergstra, Valentin Bisson, Josh Bleecher Snyder, Nicolas Bouchard, Nicolas Boulanger-Lewandowski, Xavier Bouthillier, Alexandre de Brébisson, Olivier Breuleux, Pierre-Luc Carrier, Kyunghyun Cho, Jan Chorowski, Paul Christiano, Tim Cooijmans, Marc-Alexandre Côté, Myriam Côté, Aaron Courville, Yann N. Dauphin, Olivier Delalleau, Julien Demouth, Guillaume Desjardins, Sander Dieleman, Laurent Dinh, Mélanie Ducoffe, Vincent Dumoulin, Samira Ebrahimi Kahou, Dumitru Erhan, Ziyi Fan, Orhan Firat, Mathieu Germain, Xavier Glorot, Ian Goodfellow, Matt Graham, Caglar Gulcehre, Philippe Hamel, Iban Harlouchet, Jean-Philippe Heng, Balázs Hidasi, Sina Honari, Arjun Jain, Sébastien Jean, Kai Jia, Mikhail Korobov, Vivek Kulkarni, Alex Lamb, Pascal Lamblin, Eric Larsen, César Laurent, Sean Lee, Simon Lefrancois, Simon Lemieux, Nicholas Léonard, Zhouhan Lin, Jesse A. Livezey, Cory Lorenz, Jeremiah Lowin, Qianli Ma, Pierre-Antoine Manzagol, Olivier Mastropietro, Robert T. McGibbon, Roland Memisevic, Bart van Merriënboer, Vincent Michalski, Mehdi Mirza, Alberto Orlandi, Christopher Pal, Razvan Pascanu, Mohammad Pezeshki, Colin Raffel, Daniel Renshaw, Matthew Rocklin, Adriana Romero, Markus Roth, Peter Sadowski, John Salvatier, François Savard, Jan Schlüter, John Schulman, Gabriel Schwartz, Iulian Vlad Serban, Dmitriy Serdyuk, Samira Shabanian, Étienne Simon, Sigurd Spieckermann, S. Ramana Subramanyam, Jakub Sygnowski, Jérémie Tanguay, Gijs van Tulder, Joseph Turian, Sebastian Urban, Pascal Vincent, Francesco Visin, Harm de Vries, David Warde-Farley, Dustin J. Webb, Matthew Willson, Kelvin Xu, Lijun Xue, Li Yao, Saizheng Zhang, and Ying Zhang. “*Theano: A Python framework for fast computation of mathematical expressions*“. arXiv e-prints, abs/1605.02688, May 2016.
- [13] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. “*Theano: a CPU and GPU Math Expression Compiler*“. In “*Proceedings of the Python for Scientific Computing Conference (SciPy)*“, June 2010. Oral Presentation.
- [14] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. “*Theano: new features and speed improvements*“. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.
- [15] Ronan Collobert, Samy Bengio, and Johnny Mariéthoz. “*Torch: a modular machine learning software library*“. Technical report, Idiap, 2002.
- [16] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. “*Torch7: A matlab-like environment for machine learning*“. In “*BigLearn, NIPS Workshop*“, number EPFL-CONF-192376, 2011.

- [17] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. “*Implementing neural networks efficiently*”. In “*Neural Networks: Tricks of the Trade*“, pages 537–557. Springer, 2012.