



# *Image Processing*

ΥΠ. ΔΙΔΑΚΤΟΡΕΣ:  
ΧΑΣΑΠΗ ΜΑΡΙΑ-ΚΩΝΣΤΑΝΤΙΝΑ  
ΧΑΣΑΠΗ ΛΑΜΠΡΙΝΗ





# Βασικές αρχές ψηφιακής επεξεργασίας εικόνας

---

Ψηφιακή επεξεργασία εικόνας σημαίνει επεξεργασία ψηφιακής εικόνας μέσω ψηφιακού υπολογιστή. Μπορούμε επίσης να πούμε ότι πρόκειται για χρήση αλγορίθμων υπολογιστών, προκειμένου να αποκτήσουμε βελτιωμένη εικόνα για να εξαγάγουμε κάποιες χρήσιμες πληροφορίες.

Στόχος της ψηφιακής επεξεργασίας εικόνας είναι η βελτίωση της ποιότητας των εικόνων, η εξαγωγή σημαντικών πληροφοριών από τις εικόνες και η αυτοματοποίηση εργασιών που βασίζονται σε εικόνες.

# Βασικά βήματα που εμπλέκονται στην ψηφιακή επεξεργασία εικόνας



## 1. Image acquisition

Λήψη εικόνας: λήψη μιας εικόνας χρησιμοποιώντας μια ψηφιακή κάμερα ή σαρωτή ή την εισαγωγή μιας υπάρχουσας εικόνας σε έναν υπολογιστή.

## 2. Image enhancement

Βελτίωση εικόνας: βελτίωση της οπτικής ποιότητας μιας εικόνας, όπως η αύξηση της αντίθεσης, η μείωση του θορύβου.

## 3. Image restoration

Επαναφορά εικόνας: αφαίρεση της υποβάθμισης από μια εικόνα, όπως το θάμπωμα, ο θόρυβος και η παραμόρφωση.

## 4. Image segmentation

Τμηματοποίηση εικόνας: περιλαμβάνει τη διαίρεση μιας εικόνας σε περιοχές ή τμήματα, καθένα από τα οποία αντιστοιχεί σε ένα συγκεκριμένο αντικείμενο ή χαρακτηριστικό της εικόνας.

# Βασικά βήματα που εμπλέκονται στην ψηφιακή επεξεργασία εικόνας



## 5. Image representation and description

Αναπαράσταση και περιγραφή εικόνας: περιλαμβάνει την αναπαράσταση μιας εικόνας με τρόπο που μπορεί να αναλυθεί από έναν υπολογιστή, και να περιγραφούν τα χαρακτηριστικά της.

## 6. Image analysis

Ανάλυση εικόνας: περιλαμβάνει τη χρήση αλγορίθμων και μαθηματικών μοντέλων για την εξαγωγή πληροφοριών από μια εικόνα, όπως η αναγνώριση αντικειμένων, η ανίχνευση μοτίβων και ο ποσοτικός προσδιορισμός χαρακτηριστικών.

## 7. Image synthesis and compression

Σύνθεση και συμπίεση εικόνας: περιλαμβάνει τη δημιουργία νέων εικόνων ή τη συμπίεση υπάρχουσών εικόνων για τη μείωση των απαιτήσεων αποθήκευσης και μετάδοσης.

Η ψηφιακή επεξεργασία εικόνας χρησιμοποιείται ευρέως σε μια ποικιλία εφαρμογών, όπως η ιατρική απεικόνιση, η τηλεπισκόπηση, η όραση υπολογιστή και τα πολυμέσα.

# Τι περιλαμβάνει η επεξεργασία εικόνας

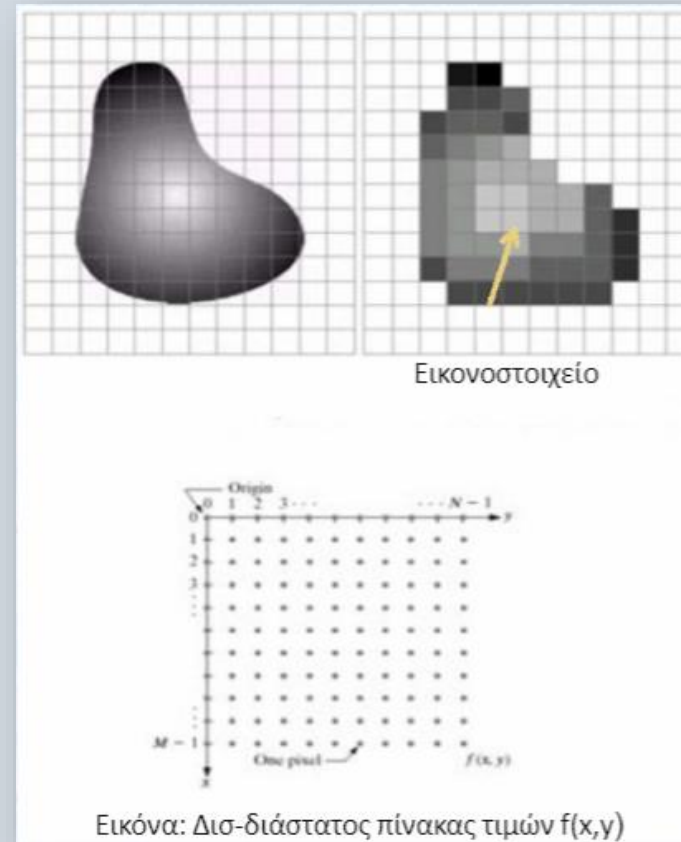


1. Εισαγωγή της εικόνας μέσω εργαλείων λήψης εικόνας.
2. Ανάλυση και χειρισμός της εικόνας.
3. Έξοδος στην οποία το αποτέλεσμα της εικόνας μπορεί να τροποποιηθεί ή να είναι μια αναφορά που βασίζεται στην ανάλυση αυτής της εικόνας και να περιγράφει την πορεία της τροποποίησής της.

# Τι είναι μια εικόνα

Μια εικόνα μπορεί να οριστεί από έναν δισδιάστατο πίνακα ειδικά διατεταγμένο σε γραμμές και στήλες. Η ψηφιακή εικόνα αποτελείται από έναν πεπερασμένο αριθμό στοιχείων, καθένα από τα οποία στοιχεία έχει μια συγκεκριμένη τιμή σε μια συγκεκριμένη θέση. Αυτά τα στοιχεία αναφέρονται ως *στοιχεία εικόνας* και *εικονοστοιχεία*.

Ένα *εικονοστοιχείο* χρησιμοποιείται ευρέως για να υποδηλώσει τα στοιχεία μιας Ψηφιακής Εικόνας.





# Τύποι εικόνας

**BINARY IMAGE:** Η δυαδική εικόνα, όπως υποδηλώνει και το όνομά της, περιέχει μόνο δύο στοιχεία pixel, δηλαδή 0 & 1, όπου το 0 αναφέρεται στο μαύρο και το 1 στο λευκό. Αυτή η εικόνα είναι επίσης γνωστή ως Μονόχρωμη.

**GREYSCALE IMAGE:** Η εικόνα σε κλίμακα του γκρι είναι μια ασπρόμαυρη εικόνα με διάφορες αποχρώσεις του γκρι.

**8 bit COLOR FORMAT:** Είναι η πιο διάσημη μορφή εικόνας. Διαθέτει 256 διαφορετικές αποχρώσεις χρωμάτων και είναι κοινώς γνωστή ως εικόνα σε κλίμακα του γκρι. Σε αυτήν τη μορφή, το 0 σημαίνει μαύρο, και το 255 σημαίνει λευκό και το 127 σημαίνει γκρι.

**16 bit COLOR FORMAT:** Είναι μια έγχρωμη μορφή εικόνας. Έχει 65.536 διαφορετικά χρώματα. Είναι επίσης γνωστή ως High Color Format. Σε αυτή τη μορφή, η κατανομή του χρώματος δεν είναι ίδια με την εικόνα σε κλίμακα του γκρι.

Μια μορφή 16 bit χωρίζεται στην πραγματικότητα σε τρεις περαιτέρω μορφές που είναι το κόκκινο, το πράσινο και το μπλε. Αυτή η περίφημη μορφή RGB.



# Φάσεις επεξεργασίας εικόνας

## ACQUISITION

### ΣΥΛΛΟΓΗ:

Θα μπορούσε να είναι τόσο απλό όσο να δίνεται μια εικόνα σε ψηφιακή μορφή. Η κύρια εργασία περιλαμβάνει:

- α) Μετατροπή εικόνας σε μορφή scale
- β) Μετατροπή χρώματος (RGB σε Γκρι ή αντίστροφα)

## IMAGE ENHANCEMENT

### ΒΕΛΤΙΩΣΗ ΕΙΚΟΝΑΣ:

Χρησιμοποιείται για την εξαγωγή ορισμένων κρυφών λεπτομερειών από μια εικόνα και είναι υποκειμενική.

## IMAGE RESTORATION

### ΑΠΟΚΑΤΑΣΤΑΣΗ ΕΙΚΟΝΑΣ:

Ασχολείται επίσης με την βελτίωση μιας εικόνας αλλά είναι αντικειμενική (Η αποκατάσταση βασίζεται σε μαθηματικό ή πιθανολογικό μοντέλο ή υποβάθμιση εικόνας).

## COLOR IMAGE PROCESSING

### ΕΠΕΞΕΡΓΑΣΙΑ ΧΡΩΜΑΤΟΣ ΕΙΚΟΝΑΣ:

Ασχολείται με την ψευδόχρωμη και την πλήρη έγχρωμη επεξεργασία εικόνας. Τα μοντέλα που προκύπτουν χρησιμοποιούνται για την ψηφιακή επεξεργασία εικόνας.

## WAVELETS AND MULTI-RESOLUTION PROCESSING

### ΚΥΜΑΤΑ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑ ΠΟΛΛΑΠΛΩΝ ΑΝΑΛΥΣΕΩΝ:

Χρησιμοποιείται για την ανάλυση ενός σήματος εικόνας σε διαφορετικές συνιστώσες συχνότητας σε διαφορετικές κλίμακες ανάλυσης (δλδ πολλαπλή ανάλυση)





# Φάσεις επεξεργασίας εικόνας

## **IMAGE COMPRESSION**

### **ΣΥΜΠΙΕΣΗ ΕΙΚΟΝΑΣ:**

Περιλαμβάνει την ανάπτυξη ορισμένων λειτουργιών για την εκτέλεση της συμπίεσης η οποία ασχολείται κυρίως με το μέγεθος ή την ανάλυση της εικόνας.

## **MORPHOLOGICAL PROCESSING**

### **ΜΟΡΦΟΛΟΓΙΚΗ ΕΠΕΞΕΡΓΑΣΙΑ:**

Ασχολείται με εργαλεία εξαγωγής στοιχείων εικόνας που είναι χρήσιμα στην αναπαράσταση & περιγραφή του σχήματος.

## **SEGMENTATION PROCEDURE**

### **ΔΙΑΔΙΚΑΣΙΑ ΤΜΗΜΑΤΟΠΟΙΗΣΗΣ:**

Περιλαμβάνει την κατάτμηση μιας εικόνας στα συστατικά μέρη ή αντικείμενά της. Η αυτόνομη τμηματοποίηση είναι η πιο δύσκολη εργασία στην Επεξεργασία Εικόνας.

## **REPRESENTATION & DESCRIPTION**

### **ΑΝΑΠΑΡΑΣΤΑΣΗ & ΠΕΡΙΓΡΑΦΗ:**

Ακολουθεί την έξοδο του σταδίου τμηματοποίησης, επιλέγοντας μια αναπαράσταση η οποία είναι το βασικό στάδιο για τη μετατροπή των πρωτογενών δεδομένων σε επεξεργασμένα δεδομένα.

## **OBJECT DETECTION AND RECOGNITION**

### **ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΑΝΑΓΝΩΡΙΣΗ ΑΝΤΙΚΕΙΜΕΝΟΥ:**

Είναι μια διαδικασία που εκχωρεί μια ετικέτα σε ένα αντικείμενο με βάση την περιγραφή του.

	INPUT	OUTPUT
IMAGE	DIGITAL IMAGE PROCESSING 1	COMPUTER VISION 2
DESCRIPTION	COMPUTER GRAPHICS 3	ARTIFICIAL INTELLIGENCE 4

Επικαλυπτόμενα  
πεδία με  
επεξεργασία  
εικόνας

**Σύμφωνα με το μπλοκ 1**, εάν η είσοδος είναι μια εικόνα και έχουμε και ως έξοδο πάλι μια εικόνα, τότε ονομάζεται Ψηφιακή Επεξεργασία Εικόνας.

**Σύμφωνα με το μπλοκ 2**, εάν η είσοδος είναι μια εικόνα και λαμβάνουμε κάποιο είδος πληροφοριών ή περιγραφής ως έξοδο, τότε ονομάζεται Υπολογιστική Όραση.

**Σύμφωνα με το μπλοκ 3**, εάν η είσοδος είναι κάποια περιγραφή ή κώδικας και λαμβάνουμε εικόνα ως έξοδο, τότε ονομάζεται Γραφικά Υπολογιστών.

**Σύμφωνα με το μπλοκ 4**, εάν η είσοδος είναι περιγραφή ή ορισμένες λέξεις-κλειδιά ή κάποιος κώδικας και λάβουμε περιγραφή ή κάποιες λέξεις-κλειδιά ως έξοδο, τότε ονομάζεται Τεχνητή Νοημοσύνη



# Πλεονεκτήματα της ψηφιακής επεξεργασίας εικόνας

**Βελτιωμένη ποιότητα εικόνας:** Οι αλγόριθμοι ψηφιακής επεξεργασίας εικόνας μπορούν να βελτιώσουν την οπτική ποιότητα των εικόνων, καθιστώντας τις πιο καθαρές, ευκρινέστερες και με την δυνατότητα να μας δίνουν περισσότερη πληροφορία.

**Αυτοματοποιημένες εργασίες που βασίζονται σε εικόνα:** Η ψηφιακή επεξεργασία εικόνας μπορεί να αυτοματοποιήσει πολλές εργασίες που βασίζονται σε εικόνες, όπως η αναγνώριση αντικειμένων, η ανίχνευση μοτίβων και η ορισμός συντεταγμένων.

**Αυξημένη απόδοση:** Οι αλγόριθμοι επεξεργασίας ψηφιακών εικόνων μπορούν να επεξεργάζονται εικόνες πολύ πιο γρήγορα από τους ανθρώπους, καθιστώντας δυνατή την ανάλυση μεγάλων ποσοτήτων δεδομένων σε σύντομο χρονικό διάστημα.

**Αυξημένη ακρίβεια:** Οι αλγόριθμοι επεξεργασίας ψηφιακών εικόνων μπορούν να παρέχουν πιο ακριβή αποτελέσματα από τους ανθρώπους, ειδικά για εργασίες που απαιτούν ακριβείς μετρήσεις ή ποσοτική ανάλυση.



# Μειονεκτήματα της ψηφιακής επεξεργασίας εικόνας

**Υψηλό υπολογιστικό κόστος:** Ορισμένοι αλγόριθμοι επεξεργασίας ψηφιακής εικόνας απαιτούν σημαντικούς υπολογιστικούς πόρους.

**Περιορισμένη ερμηνευτικότητα:** Ορισμένοι αλγόριθμοι επεξεργασίας ψηφιακής εικόνας μπορεί να παράγουν αποτελέσματα που είναι δύσκολο να ερμηνευτούν από τον άνθρωπο, ειδικά για πολύπλοκους ή εξελιγμένους αλγόριθμους.

**Εξάρτηση από την ποιότητα της εισαγωγής:** Η ποιότητα της εξόδου των αλγορίθμων επεξεργασίας ψηφιακής εικόνας εξαρτάται σε μεγάλο βαθμό από την ποιότητα των εικόνων εισόδου. Οι εικόνες εισόδου κακής ποιότητας μπορεί να έχουν ως αποτέλεσμα κακή ποιότητα εξόδου.

**Περιορισμοί αλγορίθμων:** Οι αλγόριθμοι επεξεργασίας ψηφιακής εικόνας έχουν περιορισμούς, όπως η δυσκολία αναγνώρισης αντικειμένων σε ακατάστατες ή κακώς φωτισμένες σκηνές ή η αδυναμία αναγνώρισης αντικειμένων με σημαντικές παραμορφώσεις.

**Εξάρτηση από καλά δεδομένα εκπαίδευσης:** Η απόδοση πολλών αλγορίθμων επεξεργασίας ψηφιακής εικόνας εξαρτάται από την ποιότητα των δεδομένων εκπαίδευσης που χρησιμοποιούνται για την ανάπτυξη των αλγορίθμων. Τα δεδομένα εκπαίδευσης κακής ποιότητας μπορούν να οδηγήσουν σε κακή απόδοση του αλγόριθμου.



# TensorFlow





# TENSORFLOW

---

Η TensorFlow , ένα μια πλήρης βιβλιοθήκη λογισμικού open source (ανοικτού κώδικα), ειδικά διαμορφωμένη για την δημιουργία εφαρμογών Μηχανικής και Βαθιάς Μάθησης. Διαθέτει μια ολοκληρωμένη και ευέλικτη σειρά εργαλείων, διευκόλυνσης στην δημιουργία και εκτέλεση τέτοιων εφαρμογών.

Οι τρεις βασικοί πυλώνες της βιβλιοθήκης TensorFlow είναι:

Η προ επεξεργασία των δεδομένων,

Η δημιουργία του μοντέλου,

Η εκτέλεση, εκπαίδευση και η εκτίμηση των αποτελεσμάτων του εκπαιδευμένου συστήματος.

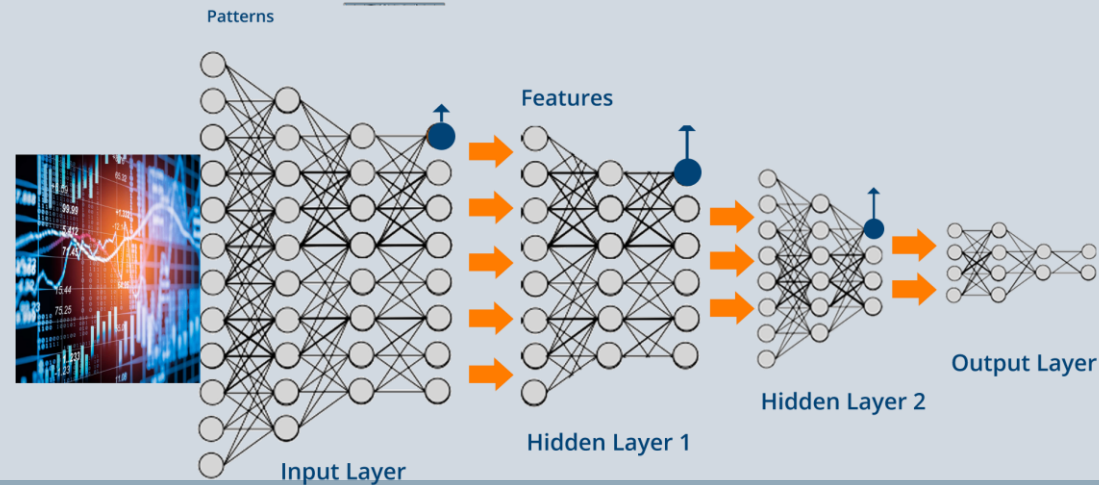


# TENSORFLOW

Η TensorFlow, παίρνει το όνομα της από ένα από τα δομικά της στοιχεία, τους τανυστές. Ο τανυστής, είναι ένα διάνυσμα ή μήτρα  $n$  διαστάσεων, το οποίο αντιπροσωπεύει κάθε είδους δεδομένα.

Ένας τανυστής, μπορεί να προκύψει είτε από τα δεδομένα εισόδου είτε από την έξοδο ενός υπολογισμού. Όλες οι εργασίες στην TensorFlow πραγματοποιούνται μέσα σε ένα Graph (γράφημα). Ο Γράφος ροής αποτυπώνει, μια σειρά υπολογισμών που συμβαίνουν με τη σειρά. Κάθε λειτουργία αναφέρεται ως κόμβος `op` και είναι όλοι συνδεδεμένοι μεταξύ τους.

Ο Γράφος, απεικονίζει τις λειτουργίες και τις σχέσεις που υπάρχουν μεταξύ των κόμβων αυτών, ωστόσο δεν απεικονίζει και τις τιμές του κάθε κόμβου. Ο τανυστής αποτελεί στην ουσία την άκρη των κόμβων.





# TENSORFLOW

---

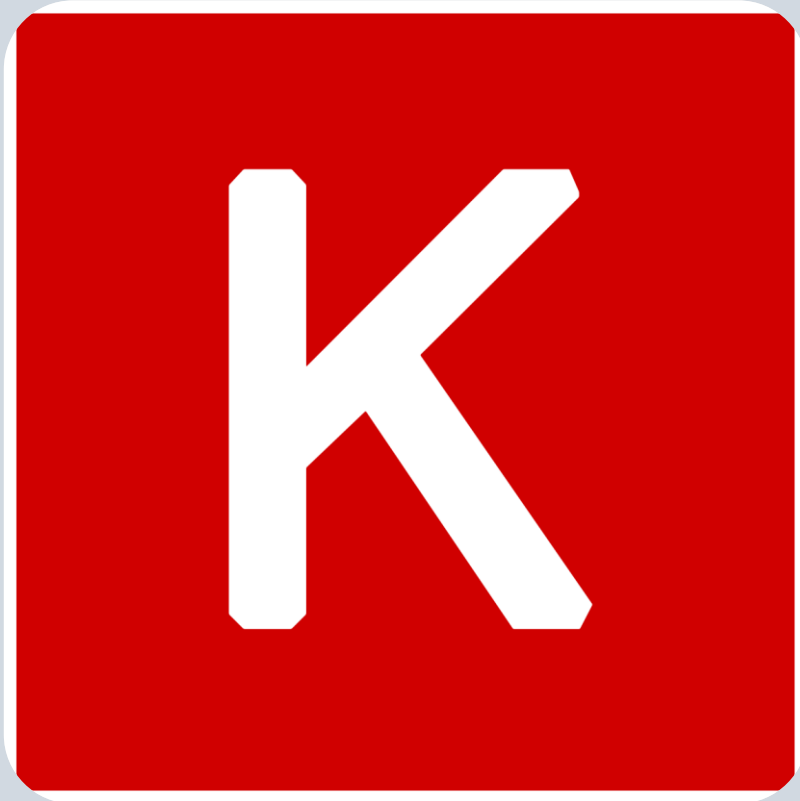
Η TensorFlow, όπως προαναφέρθηκε, αποτελεί την πιο διαδεδομένη βιβλιοθήκη λογισμικού Τεχνητής Νοημοσύνης, καθώς έχει σχεδιαστεί για να είναι φιλική προς το χρήστη.

Περιλαμβάνει μια ποικιλία API (διεπαφή προγραμματισμού εφαρμογών) για τη δημιουργία μεγάλης κλίμακας αρχιτεκτονικών βαθιάς μάθησης, όπως CNNs και RNNs. Διαθέτει επίσης, ένα εξαιρετικό εργαλείο Debugging, ενώ το γεγονός ότι έχει την ικανότητα να τρέχει μεγάλης κλίμακας εφαρμογές τόσο σε CPU όσο και σε GPU το καθιστά μια ολοκληρωτική λύση για την ανάπτυξη εφαρμογών Μηχανικής Μάθησης.





# Keras



Το Keras είναι ένα API σχεδιασμένο για ανθρώπους και όχι για μηχανές.

Ελαχιστοποιεί τον αριθμό των ενεργειών χρήστη που απαιτούνται για περιπτώσεις κοινής χρήσης και παρέχει σαφή και εφαρμόσιμα μηνύματα σφάλματος.

Δίνει την ύψιστη προτεραιότητα στη δημιουργία εξαιρετικής τεκμηρίωσης και οδηγών προγραμματιστών.

# Μια υπερδύναμη για προγραμματιστές

Ο σκοπός του Keras είναι να δώσει το πλεονέκτημα σε κάθε προγραμματιστή που θέλει να δημιουργήσει εφαρμογές που υποστηρίζονται από τη μηχανική εκμάθηση.

Το Keras εστιάζει στην ταχύτητα εντοπισμού σφαλμάτων, την κομψότητα και τη συνοπτικότητα του κώδικα, τη δυνατότητα συντήρησης και την ανάπτυξη.

Τα μοντέλα εκτελούνται πιο γρήγορα χάρη στις βελτιστοποιήσεις μεταγλώττισης XLA και Autograph και είναι πιο εύκολο να αναπτυχθούν σε κάθε επιφάνεια (διακομιστής, κινητό, πρόγραμμα περιήγησης, ενσωματωμένο) χάρη στο TF Serving, το TF Lite και το TF.js.



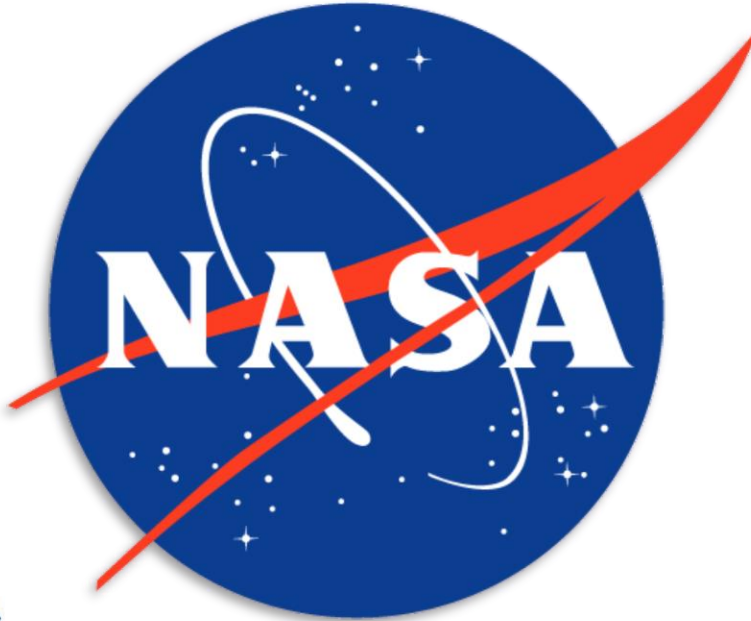
# Μηχανική εκμάθηση

Το Keras είναι ένα κεντρικό μέρος, του στενά συνδεδεμένου οικοσυστήματος TensorFlow, βασισμένο στην συγκεκριμένη πλατφόρμα, που καλύπτει κάθε βήμα της ροής εργασιών μηχανικής εκμάθησης, από τη διαχείριση δεδομένων και την εκπαίδευση παραμέτρων έως τις λύσεις ανάπτυξης.



# MACHINE LEARNING





## Έρευνα τελευταίας τεχνολογίας



Το Keras χρησιμοποιείται από το CERN, τη NASA, το NIH και πολλούς ακόμη επιστημονικούς οργανισμούς σε όλο τον κόσμο.

Προσφέρει λειτουργίες υψηλού επιπέδου για την επιτάχυνση των κύκλων πειραματισμού καθώς επίσης και ευελιξία για την εφαρμογή ερευνητικών ιδεών.



---

*Ας γνωρίσουμε λίγο καλύτερα το Keras...*





# Keras Introduction

---

Το Keras είναι μια βιβλιοθήκη νευρωνικών δικτύων ανοιχτού κώδικα γραμμένη σε γλώσσα προγραμματισμού Python που λειτουργεί παράλληλα με την TensorFlow. Έχει σχεδιαστεί για να είναι ευέλικτη, γρήγορη και εύκολη στη χρήση. Αναπτύχθηκε από τον François Chollet, μηχανικό της Google.

Δεν χειρίζεται υπολογισμούς χαμηλού επιπέδου. Αντ' αυτού, χρησιμοποιεί μια άλλη βιβλιοθήκη η οποία πραγματοποιεί αυτήν τη λειτουργία, που ονομάζεται TensorFlow και αποτελεί Backend engine.



# Keras Introduction

---

Είναι ένα API υψηλού επιπέδου, που αγκαλιάζει ή περιτυλίγει το αντίστοιχο API, του χαμηλού επιπέδου (TensorFlow).

Το API υψηλού επιπέδου (Keras) χειρίζεται τον τρόπο με τον οποίο φτιάχνουμε μοντέλα, ορίζουμε επίπεδα ή πολλαπλά μοντέλα εξόδου-εισόδου.

Συντάσσει το μοντέλο προς επεξεργασία, ενώ παρέχει και τις κατάλληλες συναρτήσεις βελτιστοποίησης καθώς και την συνάρτηση προσαρμογής του μοντέλου (Fit) κατά τη διαδικασία εκπαίδευσης.





# Keras Introduction

---

Μερικά από τα πλεονεκτήματα της βιβλιοθήκης αυτής είναι:

1. Γρήγορη και εύκολη δημιουργία νευρωνικών συστημάτων.
2. Πληθώρα παροχής συμβουλών και πρακτικών παραδειγμάτων.
3. Υποστήριξη πολλών και διαφορετικών GPUs
4. Δυνατότητα σύμπραξης με διαφορετικές Backend μηχανές , όπως η TensorFlow, η Theano και η CNTK.



# Keras Introduction

---

Το Keras είναι γραμμένο σε Python, ενώ τρέχει πάνω από την πλατφόρμα μηχανικής εκμάθησης Tensorflow.

Αναπτύχθηκε προκειμένου να παρέχει δυνατότητα γρήγορου πειραματισμού:

*«Το να μπορείτε να πηγαίνετε από μια ιδέα σε ένα αποτέλεσμα όσο το δυνατόν γρηγορότερα είναι το κλειδί για να κάνετε καλή έρευνα».*



# Keras Introduction

---

Το Keras είναι:

- **Απλό**, αλλά όχι απλοϊκό: μειώνει το γνωστικό φόρτο των προγραμματιστών για να τους απελευθερώσει να εστιάσουν στα μέρη του προβλήματος που έχουν πραγματικά σημασία.
- **Ευέλικτο**: υιοθετεί την αρχή της προοδευτικής αποκάλυψης της πολυπλοκότητας: οι απλές ροές εργασίας πρέπει να είναι γρήγορες και εύκολες, ενώ οι αυθαίρετα προηγμένες ροές εργασίας θα πρέπει να είναι δυνατές μέσω μιας σαφούς διαδρομής που βασίζεται σε όσα έχουν ήδη μάθει.
- **Ισχυρό**: παρέχει απόδοση και επεκτασιμότητα σε ισχύ στη βιομηχανία: χρησιμοποιείται από οργανισμούς και εταιρείες όπως η NASA, το YouTube ή η Waymo.



# Keras & TensorFlow 2

---

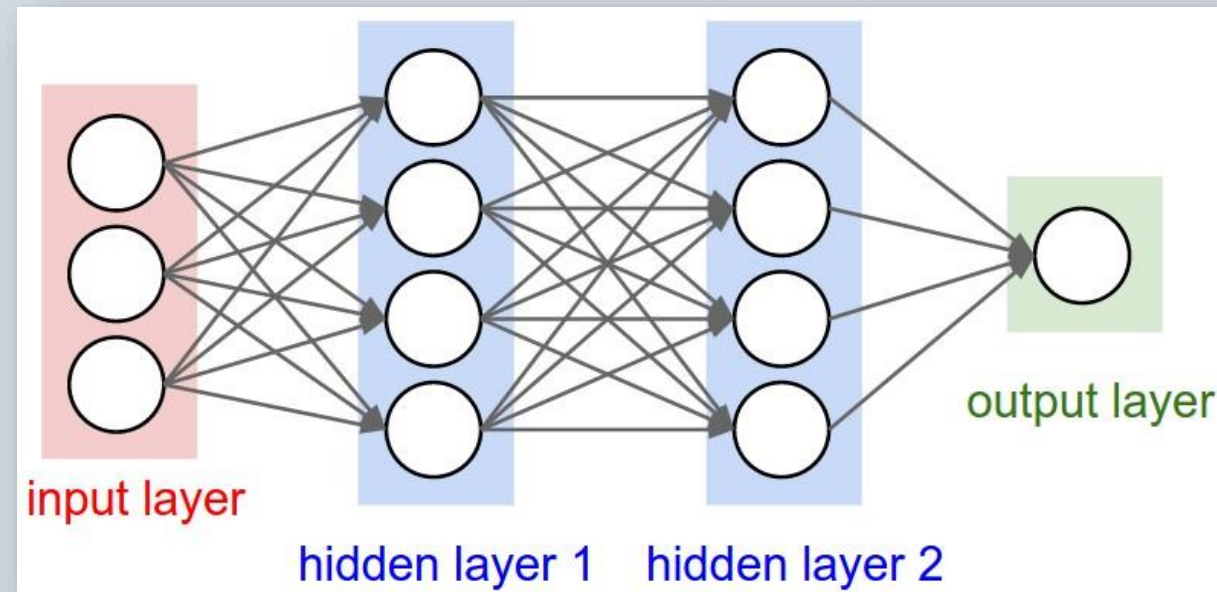
Το Keras είναι το API υψηλού επιπέδου της πλατφόρμας TensorFlow: μια προσιτή, εξαιρετικά παραγωγική διεπαφή για την επίλυση προβλημάτων μηχανικής μάθησης, με έμφαση στη σύγχρονη βαθιά μάθηση. Παρέχει βασικά δομικά στοιχεία για την εξαγωγή συμπερασμάτων μηχανικής εκμάθησης με υψηλή ταχύτητα επανάληψης.

Το Keras δίνει τη δυνατότητα στους μηχανικούς και τους ερευνητές να επωφεληθούν πλήρως από τις δυνατότητες επεκτασιμότητας και των πολλαπλών βιβλιοθηκών της πλατφόρμας TensorFlow: μπορείτε να εκτελέσετε το Keras σε TPU ή σε μεγάλα συμπλέγματα GPU και μπορείτε να εξάγετε τα μοντέλα Keras για εκτέλεση στο πρόγραμμα περιήγησης ή σε κινητή συσκευή.

# Keras Layers API

Τα επίπεδα είναι τα βασικά δομικά στοιχεία των νευρωνικών δικτύων στο Keras.

Ένα επίπεδο αποτελείται από μια λειτουργία υπολογισμού tensor-in tensor-out (μέθοδος κλήσης του επιπέδου) και κάποια κατάσταση, που σχετίζεται με τις μεταβλητές TensorFlow (τα βάρη του επιπέδου).





# Keras Layers API

---

Οι βασικές δομές δεδομένων του Keras είναι τα **επίπεδα** και τα **μοντέλα** .

Υπάρχουν τρεις τρόποι δημιουργίας μοντέλων Keras:

**To Sequential μοντέλο**, το οποίο είναι πολύ απλό (ένας απλός κατάλογος επιπέδων), αλλά περιορίζεται σε στοίβες επιπέδων μιας εισόδου και μίας εξόδου (όπως περιγράφει και το όνομά του).

**To functional API**, το οποίο είναι ένα εύχρηστο, πλήρως εξοπλισμένο API που υποστηρίζει αυθαίρετες αρχιτεκτονικές μοντέλων. Για τους περισσότερους ανθρώπους και τις περισσότερες περιπτώσεις χρήσης, αυτό χρησιμοποιείται. Αυτό είναι το μοντέλο Keras "industry strength".

**To μοντέλο subclassing**, στο οποίο εφαρμόζουμε τα πάντα από την αρχή μόνοι μας. Χρησιμοποιείται αυτό σε περίπλοκες, εξαντλημένες περιπτώσεις χρήσης έρευνας.



# Keras Layers API

---

Το Keras ακολουθεί την αρχή της **προοδευτικής αποκάλυψης της πολυπλοκότητας** :  
διευκολύνει το ξεκίνημα, ωστόσο καθιστά δυνατό τον χειρισμό αυθαίρετων προηγμένων περιπτώσεων χρήσης, που απαιτούν μόνο σταδιακή μάθηση σε κάθε βήμα.



Εδώ είναι το `Sequential` μοντέλο:

```
from tensorflow.keras.models import Sequential

model = Sequential()
```

Η στοίβαξη στρωμάτων είναι τόσο εύκολη όσο `.add()`:

```
from tensorflow.keras.layers import Dense

model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
```

# Ο πιο στοιχειώδης τρόπος χρήσης του Keras

Μόλις το μοντέλο σας φαίνεται καλό, διαμορφώστε τη διαδικασία εκμάθησής του με `.compile()`:

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

```
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.SGD(learning_rate=0.01, momentum=0.9, nesterov=True))
```

Τώρα μπορείτε να επαναλάβετε τα δεδομένα προπόνησής σας σε παρτίδες:

```
# x_train and y_train are Numpy arrays
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

Αξιολογήστε την απώλεια δοκιμής και τις μετρήσεις σας σε μία γραμμή:

```
loss_and_metrics = model.evaluate(x_test, y_test, batch_size=128)
```

Η δημιουργήστε προβλέψεις για νέα δεδομένα:

```
classes = model.predict(x_test, batch_size=128)
```





---

# ΕΝΑ ΑΠΛΟ ΠΑΡΑΔΕΙΓΜΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ ΣΤΟ ΚΕΡΑΣ



Θα καλύψουμε τα ακόλουθα σημεία σε αυτό το παράδειγμα:

---

- Φόρτωση εικόνας
- Επεξεργασία εικόνας
- Μετατροπή εικόνας σε πίνακα και αντίστροφα
- Αλλαγή χρώματος της εικόνας
- Επεξεργασία δεδομένων εικόνας



# Φόρτωση Εικόνας

---

Στο Keras, η συνάρτηση `load_img()` χρησιμοποιείται για τη φόρτωση της εικόνας.

Η εικόνα που φορτώθηκε χρησιμοποιώντας τη μέθοδο `load_img()` είναι αντικείμενο

PIL. Ορισμένες πληροφορίες δίνονται από τις εικόνες, όπως ο τύπος εικόνας που είναι

αντικείμενο PIL, η μορφή που είναι JPEG, το μέγεθος που είναι (6000,4000), η λειτουργία που

είναι RGB κλπ.

# Φόρτωση Εικόνας



Python3



```
import keras
```

```
from keras.preprocessing.image import load_img
```



```
# load the image
```

```
img = load_img('dog.jpg')
```

```
# find more about the image
```

```
print(type(img))
```

```
print(img.format)
```

```
print(img.mode)
```

```
print(img.size)
```

```
# show the image
```

```
img.show()
```

Παραγωγή:

```
<class 'PIL.JpegImagePlugin.JpegImageFile'>  
JPEG  
RGB  
(6000, 4000)
```



# Αλλαγή μεγέθους εικόνας

---

Μπορούμε να εκτελέσουμε ορισμένες λειτουργίες στην εικόνα, όπως αλλαγή μεγέθους, αλλαγή χρώματος, μετατροπή σε πίνακα κλπ. πριν εκπαιδύσουμε οποιοδήποτε μοντέλο.

Για να αλλάξετε το μέγεθος του σχήματος της εικόνας, η μέθοδος `resize()` καλείται για την εικόνα. Το μέγεθος στο οποίο θέλουμε να μετατρέψουμε την εικόνα πρέπει να είναι επαναλαμβανόμενο.

# Αλλαγή μεγέθους εικόνας

Python3

```
img = load_img('dog.jpg')  
  
# change image size  
image = img.resize([30, 30])  
  
# print new image size  
print(image.size)
```

Παραγωγή:

```
(30, 30)
```

Μπορούμε να δούμε στην έξοδο, 30 x 30 είναι το νέο μέγεθος της εικόνας.



# Μετατροπή εικόνας σε πίνακα

---

Υπάρχει η μέθοδος `img_to_array()` για τη μετατροπή εικόνων σε πίνακα και η μέθοδος `array_to_img()` για τη μετατροπή του πίνακα εικόνων σε εικόνα.

Στο παράδειγμα, μόλις έχουμε πρόσβαση στο ευρετήριο του πίνακα εικόνων. Μπορούμε να λάβουμε πληροφορίες όπως σχήμα πίνακα εικόνων, τύπος, κλπ. Όταν ο πίνακας εικόνων μετατρέπεται σε εικόνα, είναι και πάλι αντικείμενο PIL.



# Μετατροπή εικόνας σε πίνακα

---

```
# convert image into array
from keras.preprocessing.image import img_to_array
from keras.preprocessing.image import array_to_img

# convert to numpy array
img_array = img_to_array(img)
print(img_array[0])
print(img_array.dtype)
print(img_array.shape)

# convert back to image
img = array_to_img(img_array)
print(type(img))
```

## Παραγωγή:

```
[[244. 244. 244.]
 [244. 244. 244.]
 [244. 244. 244.]
 ...
 [117.  74.  42.]
 [115.  71.  42.]
 [112.  68.  39.]]
float32
(4000, 6000, 3)
<class 'PIL.JpegImagePlugin.JpegImageFile'>
```



# Αλλαγή στο χρώμα της Εικόνας

---

Για να μετατρέψετε τις έγχρωμες εικόνες σε κλίμακα του γκρι, απλώς ορίζουμε την κλίμακα `grayscale = True` στη μέθοδο `load_img()`.

Για να αποθηκεύσουμε τις εικόνες που έχουν μετατραπεί πρέπει να μετατρέψουμε την εικόνα σε πίνακα, διαφορετικά η μέθοδος `save_img()` θα προκαλέσει σφάλμα.

Μετά τη μετατροπή της έγχρωμης εικόνας σε κλίμακα του γκρι, βλέπουμε ότι εξακολουθεί να δείχνει τη λειτουργία RGB και το μέγεθος είναι το ίδιο.

# Αλλαγή στο χρώμα της Εικόνας



Python3

```
from keras.preprocessing.image import save_img
from keras.preprocessing.image import img_to_array

# load image as grayscale
img = load_img('dog.jpg', grayscale=True)

# convert image to a numpy array
img_array = img_to_array(img)

# save the image with a new filename
save_img('dog_grayscale.jpg', img_array)

# load the image to confirm it was saved
# correctly
img = load_img('dog_grayscale.jpg')
print(type(img))
print(img.format)
print(img.mode)
print(img.size)
img.show()
```

Παραγωγή:

```
<class 'PIL.Image.Image'>
None
RGB
(6000, 4000)
```



# Επεξεργασία ενός συνόλου δεδομένων εικόνας

---

Για να φορτώσετε το σύνολο δεδομένων της εικόνας, η απλή μέθοδος είναι να χρησιμοποιήσετε `load_data()` στο σύνολο δεδομένων εικόνας.

Χρησιμοποιούμε σύνολο δεδομένων `mnist` που είναι ήδη διαθέσιμο στο `Keras`. Θα δώσει σε αντάλλαγμα `x_train`, `y_train`, `x_test` και `y_test`. Τα `x_train` και `y_train` θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου και τα `x_test` και `y_test` θα χρησιμοποιηθούν για σκοπούς δοκιμής.

Μπορούμε να αναδιαμορφώσουμε όλες τις εικόνες μέσα στο σύνολο δεδομένων χρησιμοποιώντας τη μέθοδο `reshape()` και να ορίσουμε ποιος τύπος εικόνων θα πρέπει να είναι όπως `'float64'` ή `'float32'`.



# Επεξεργασία ενός συνόλου δεδομένων εικόνας

60000 εικόνες στο mnist έχουν αναδιαμορφωθεί σε μέγεθος 28 x 28 και οι εικόνες είναι τύπου numpy.ndarray

Python3



```
from keras.datasets import mnist
```



```
(X_train, Y_train), (X_test, Y_test) = mnist.load_data()
```

```
# reshape the image
```

```
images = X_train.reshape(-1, 28, 28, 1).astype('float64')
```

```
print(images.shape)
```

```
print(type(images))
```

```
print(images.size)
```

Παραγωγή:

```
(60000, 28, 28, 1)
<class 'numpy.ndarray'>
47040000
```



---

# ΕΝΑ ΠΑΡΑΔΕΙΓΜΑ ΕΠΕΞΕΡΓΑΣΙΑΣ ΙΑΤΡΙΚΗΣ ΕΙΚΟΝΑΣ ΣΤΟ ΚΕΡΑΣ



# Ανάλυση Ιατρικών Εικόνων Υπερήχου Θυρεοειδούς με Χρήση Τεχνικών Βαθιάς Μάθησης

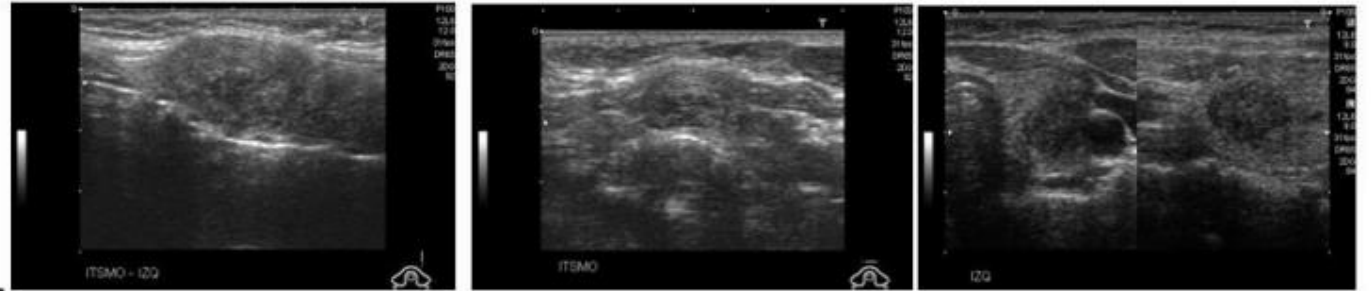
Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα εργασία περιλαμβάνει εικόνες υπερήχων θυρεοειδούς (αρχεία jpg) από 383 ασθενείς (462 εικόνες). Σε κάθε ασθενή αντιστοιχούν μία ή περισσότερες εικόνες υπερήχου (RGB), μεγέθους 560x360 pixel. Οι πληροφορίες για κάθε ασθενή όπως η ηλικία, το φύλο, ο αριθμός των εικόνων υπερήχου που του αντιστοιχούν, ο βαθμός κινδύνου κακοήθειας, το πλήθος, οι συντεταγμένες, η σύσταση και η ηχογένεια των όζων, βρίσκονται σε ένα αρχείο xml.





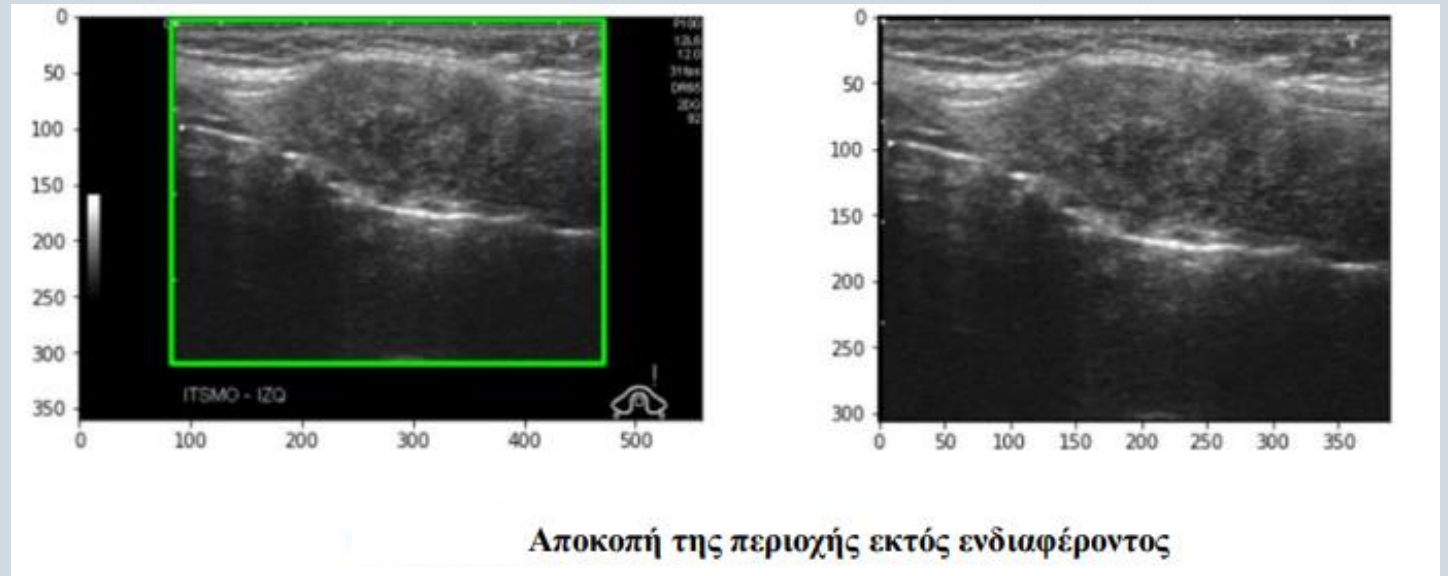
# Τμηματοποίηση εικόνων υπερήχου θυρεοειδούς

Προεπεξεργασία  
Συνόλου Δεδομένων  
(Dataset Preprocessing)



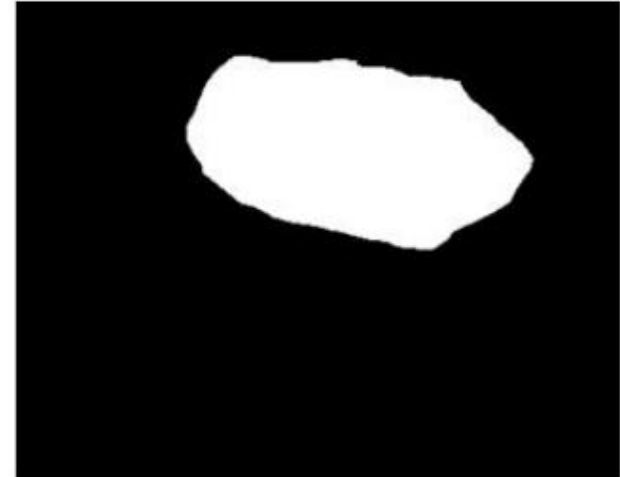
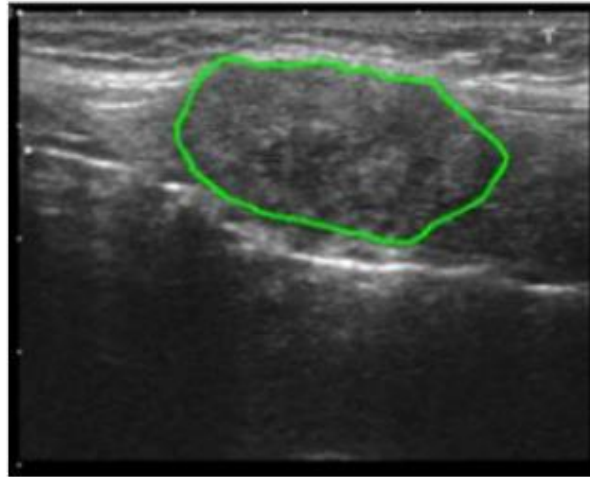
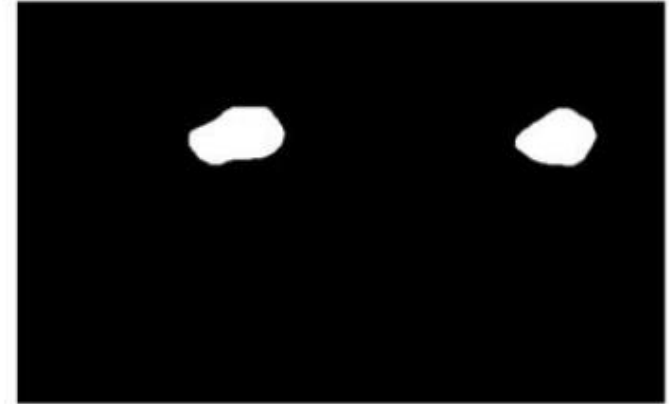
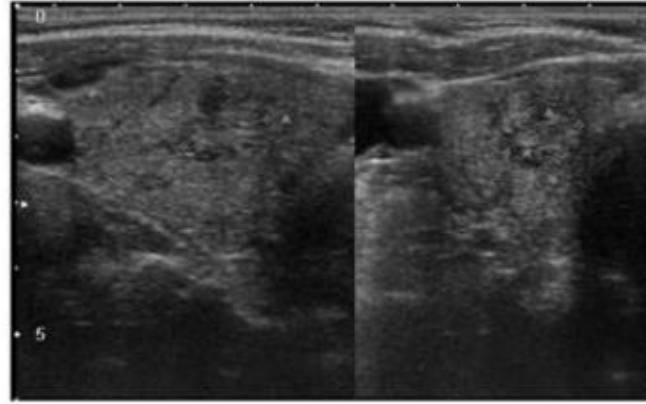
Εικόνες υπερήχου όζων θυρεοειδούς από το σύνολο δεδομένων

# Αποκοπή της περιοχής εκτός ενδιαφέροντος



# Δημιουργία Ground Truth εικόνων

Δημιουργούμε τις αντίστοιχες ground truth εικόνες, όπου η περιοχή του υποβάθρου είναι μαύρη (label 0) και η περιοχή του ή των όζων λευκή (label 1)



**Εικόνα 41: Εικόνες υπερήχου θυρεοειδούς και η αντίστοιχη ground truth εικόνα**



1.Φόρτωση εικόνων

2.Διαχωρισμός του  
συνόλου δεδομένων

3.Κατασκευή και  
εκπαίδευση μοντέλου

Εκπαίδευση  
νευρωνικού  
δικτύου

# Αξιολόγηση της εκπαίδευσης του μοντέλου

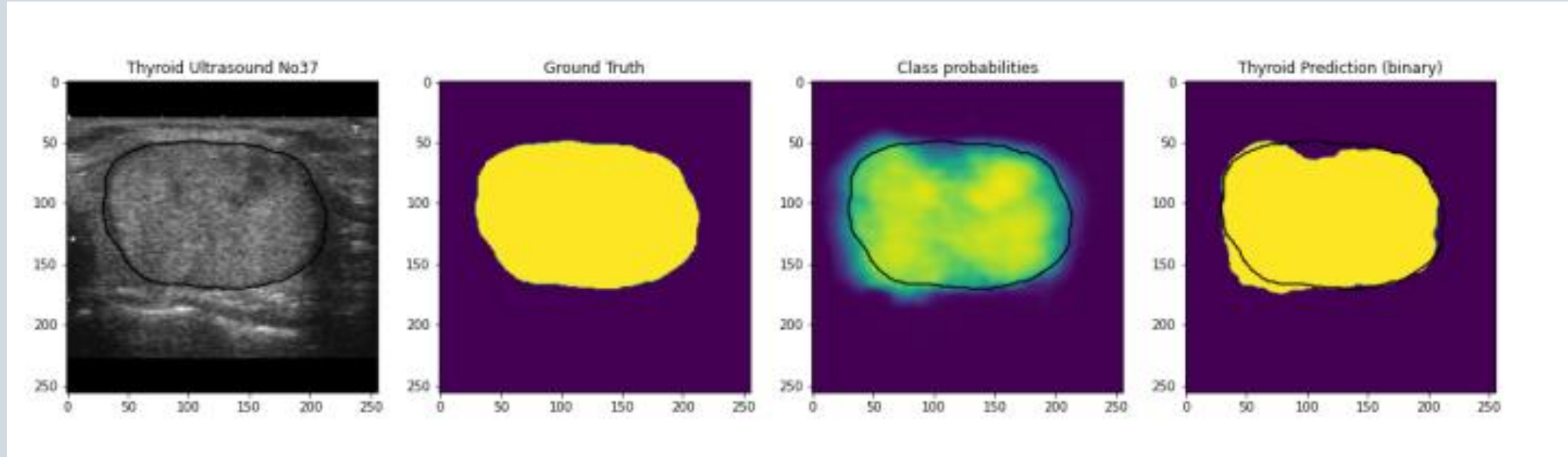
Οι μετρικές που χρησιμοποιούμε για την αξιολόγηση του αποτελέσματος της εκπαίδευσης

του μοντέλου είναι:

- Accuracy
- mean Intersection over Union (mIoU)
- Dice Similarity Coefficient ή F1 score
- Recall
- Precision

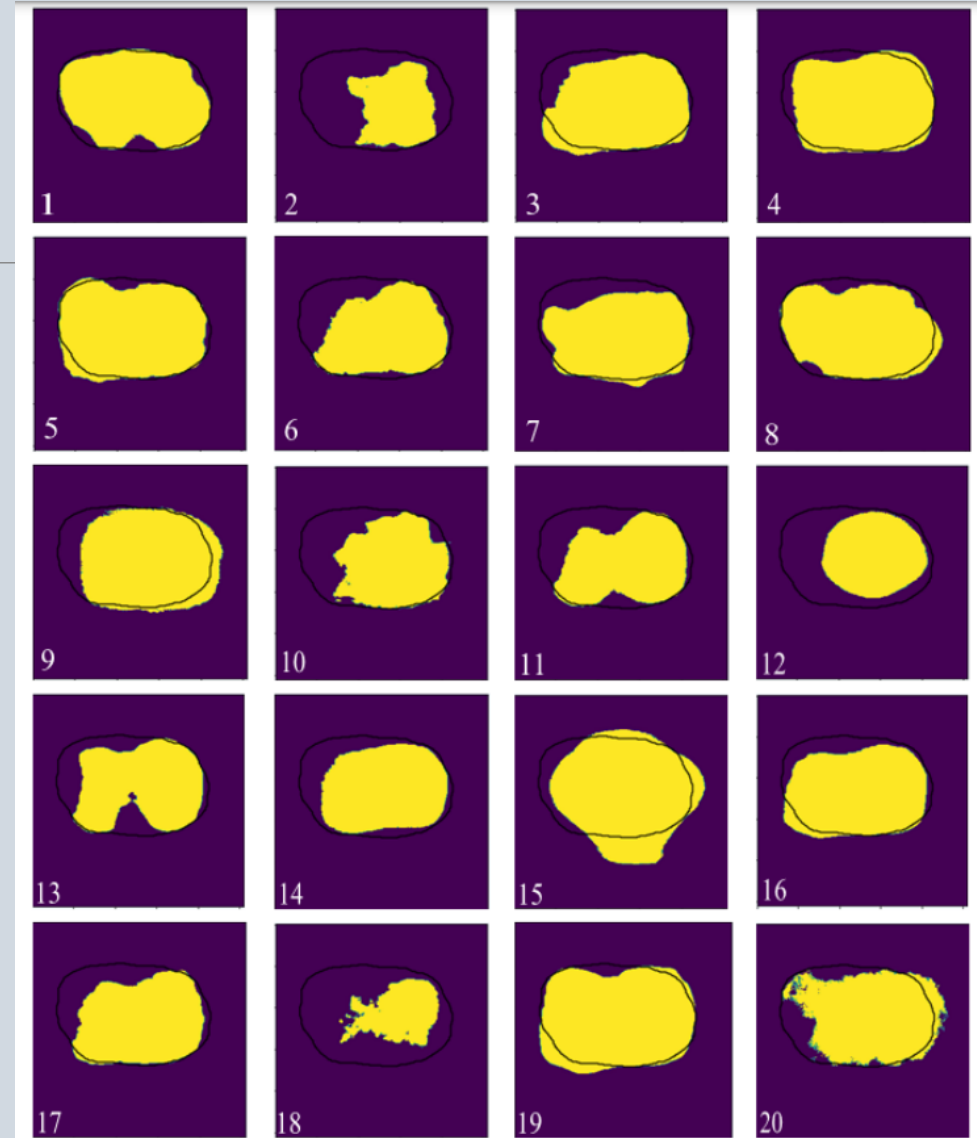
	L	BN	DO	LR	BS	#Filters	Opt.	#Params	Loss	Acc.	mIoU	Dice	Recall	Precision
1	6	NAI	OXI	0.001	32	32	Adam	34.5M	0.1847	0.9304	0.7559	0.6712	0.6690	0.7642
2	6	NAI	OXI	0.001	16	32	Adam	34.5M	0.1679	0.9357	0.7401	0.6596	0.6262	0.8357
3	6	NAI	OXI	0.001	8	32	Adam	34.5M	0.1416	0.9417	0.7869	0.7303	0.7379	0.7968
4	5	NAI	OXI	0.001	8	32	Adam	8.5M	0.1599	0.9436	0.7759	0.7288	0.6905	0.8451
5	6	NAI	OXI	0.001	4	32	Adam	34.5M	0.1405	0.9442	0.7895	0.7504	0.7296	0.8363
6	6	NAI	OXI	0.001	8	64	Adam	138M	0.1409	0.9369	0.7558	0.6653	0.6552	0.8214
7	6	NAI	OXI	0.0001	8	32	Adam	34.5M	0.2049	0.9401	0.7577	0.6323	0.6587	0.8442
8	5	NAI	OXI	0.001	16	32	Adam	8.5M	0.1797	0.9424	0.7704	0.7295	0.7380	0.7946
9	6	OXI	0.01	0.001	8	32	Adam	34.5M	0.1518	0.9334	0.7765	0.7015	0.7511	0.7389
10	6	OXI	0.50	0.001	8	32	Adam	34.5M	0.1608	0.9322	0.7643	0.6605	0.7216	0.7460
11	5	NAI	OXI	0.001	8	64	Adam	34.5M	0.1677	0.9344	0.7404	0.6533	0.6758	0.7870
12	6	NAI	0.01	0.001	8	32	Adam	34.5M	0.1774	0.9174	0.6848	0.5532	0.5070	0.7774
13	5	NAI	OXI	0.01	8	64	Adam	34.5M	0.1622	0.9326	0.7324	0.6343	0.6328	0.8049
14	6	NAI	0.01	0.01	8	32	Adam	34.5M	0.1733	0.9256	0.7045	0.6048	0.7305	0.8045
15	6	NAI	0.01	0.01	4	32	Adam	34.5M	0.2055	0.9062	0.6742	0.5280	0.5513	0.6664
16	6	NAI	0.10	0.01	8	32	Adam	34.5M	0.1555	0.9368	0.7534	0.6691	0.6682	0.8102
17	5	NAI	0.01	0.01	8	64	Adam	34.5M	0.1655	0.9281	0.7151	0.6189	0.5904	0.8019
18	6	NAI	0.01	0.01	16	32	Adam	34.5M	0.1758	0.9211	0.6793	0.5780	0.4943	0.8260
19	5	NAI	OXI	0.01	16	32	Adam	8.5M	0.2080	0.9256	0.7408	0.6863	0.7273	0.7065
20	6	NAI	0.01	0.01	8	32	SGD	34.5M	0.1951	0.9216	0.7025	0.5732	0.6460	0.7193

Αποτελέσματα μετρικών εκπαίδευσης



Τμηματοποίηση μεγάλου όζου

Τμηματοποίηση που  
αντιστοιχεί στα  
μοντέλα του πίνακα  
αποτελεσμάτων







# Βιβλιογραφία

---

Digital Image Processing (Rafael c. gonzalez)

“Digital Image Processing” by Rafael C. Gonzalez and Richard E. Woods.

“Computer Vision: Algorithms and Applications” by Richard Szeliski.

“Digital Image Processing Using MATLAB” by Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins.

Χρονάκη Γ. (2022). “Ανάλυση Ιατρικών Εικόνων Υπερήχου Θυρεοειδούς με Χρήση Τεχνικών Βαθιάς Μάθησης ”



---

Σας ευχαριστώ!