

# Αναφορά για Hyperspectral Imaging - up2metric

Νίκος Σούρλος

Νοέμβριος 2020

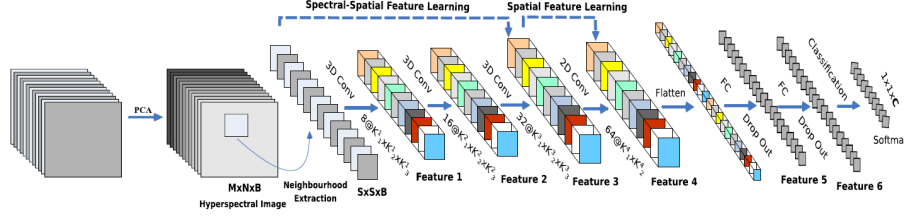
## 1 Προεπεξεργασία Δεδομένων

Η μέθοδος που υιοθετήθηκε με σκοπό την ταξινόμηση hyperspectral εικόνων είναι αυτή που παρουσιάζεται στο paper "HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification" [1]. Η τεχνική αυτή συνδυάζει ένα 3D-CNN με ένα 2D-CNN αφού πρώτα χρησιμοποιηθεί η μέθοδος PCA (Principal Component Analysis) ώστε να εξαιρεθούν προβλήματα που είναι πιθανό να υπάρξουν λόγω της ομοιότητας μεταξύ των τάξεων ταξινόμησης και των εσωτερικών διαφορών που υπάρχουν στα στοιχεία κάθε τάξης. Το PCA αφαιρεί τις πλεονάζουσες φασματικές ζώνες, μειώνοντας έτσι των συνολικό αριθμό των ζωνών που θα χρησιμοποιηθούν από τα CNN που αναφέρθηκαν παραπάνω, διατηρώντας τις χωρικές διαστάσεις αμετάβλητες. Αν αρχικά τα δεδομένα μας μπορούσαν να αναπαρασταθούν ως ένας κύβος διαστάσεων  $N_x \times N_y \times N$  όπου οι πρώτες δύο είναι οι χωρικές διαστάσεις και  $N$  ο αριθμός των φασματικών ζωνών, το αποτέλεσμα της εφαρμογής αυτής του PCA είναι ένας κύβος διαστάσεων  $N_x \times N_y \times B$  όπου  $B < N$ . Επίσης, είναι σκόπιμο να αναφερθεί ότι πριν τα δεδομένα εισαχθούν στα προαναφερθέντα δίκτυα, ο κύβος που έχει προκύψει διαιρείται σε μικρά αλληλοεπικαλυπτόμενα 3D-patches, με το ground-truth label του καθενός από αυτά να ορίζεται ως το ground-truth label του κεντρικού pixel του εκάστοτε 3D-patch. Το κάθε ένα από αυτά τα 3D-patches καλύπτει ένα παράθυρο  $S \times S$ , και όλες τις φασματικές ζώνες που περιέχουν τα pixel που περικλύονται από αυτό το παράθυρο. Τέλος, αξίζει να σημειωθεί ότι επειδή ορίσαμε το ground-truth label του κάθε παραθύρου ως αυτό του κεντρικού pixel, το  $S$  πρέπει να είναι περιττός αριθμός (ώστε να υπάρχει κεντρικό pixel).

## 2 Δομή Δικτύου και Σχεδιαστικές Αποφάσεις

Η δομή του δικτύου υιοθετήθηκε από το paper [1], προσαρμόζοντας κατάλληλα τις διαστάσεις της κάθε layer σύμφωνα με τις ανάγκες των δεδομένων εισόδου. Η αρχιτεκτονική του δικτύου φαίνεται στο Σχ.1. Πιο συγκεκριμένα, οι διαστάσεις των τριών 3D convolutional kernels που επιλέχθηκαν για τα δεδομένα μας είναι  $8 \times 2 \times 2 \times 7 \times 1$  ( $K_1^1 = 2, K_2^1 = 2, K_3^1 = 7$ ),  $16 \times 2 \times 2 \times 5 \times 1$  ( $K_1^2 = 2, K_2^2 = 2, K_3^2 = 5$ ) και  $32 \times 3 \times 3 \times 3 \times 1$  ( $K_1^3 = 3, K_2^3 = 3, K_3^3 = 3$ ) για την πρώτη, την δεύτερη και

την τρίτη διάσταση αντίστοιχα, όπου  $8 \times 2 \times 2 \times 7 \times 1$  σημαίνει ότι χρησιμοποιούνται 8 3D kernels διαστάσεων  $2 \times 2 \times 7$ . Οι διαστάσεις του 2D convolutional kernel, που έπεται αμέσως μετά από τα 3D kernels (και το reshape ώστε να έχουμε ένα 2D matrix) είναι  $64 \times 3 \times 3 \times 576$  ( $K_1^4 = 3$ ,  $K_2^4 = 3$ ), όπου χρησιμοποιούνται 64 2D kernels διαστάσεων  $3 \times 3$  και 576 είναι ο αριθμός των 2D input feature maps.



Σχήμα 1: Αρχιτεκτονική του HybridSN [1]

Το αρχικό test ratio που χρησιμοποιήθηκε στο paper είναι 0.7. Στην περίπτωση μας, παρατηρήθηκε ότι, διατηρώντας όλες τις άλλες παραμέτρους αμετάβλητες, μπορούμε να πετύχουμε μια βελτίωση της τάξης του 7-8% χρησιμοποιώντας για test ratio την τιμή 0.3 (90% με 98% αντίστοιχα). Επιπλέον, ενώ οι συγγραφείς χρησιμοποίησαν ένα παράθυρο με διαστάσεις  $25 \times 25$ , σε εμάς αυτό δεν κατέστη εφικτό λόγω του ότι η διαθέσιμη μνήμη RAM στο google colab ήταν μόνο 12GB. Για τον λόγο αυτό επιλέξαμε να χρησιμοποιήσουμε το μέγιστο δυνατό παράθυρο που αυτή επιτρέπει, το οποίο είναι  $13 \times 13$  (μεγαλύτερα παράθυρα οδηγούν σε run out of memory error).

Κατά τον ορισμό της function που δημιουργεί τα 3D-patches ("createImageCubes") όλα τα pixels των οποίων οι τιμές των ground-truth labels είναι 0 (pixels που δεν μας ενδιαφέρουν) αφαιρούνται και δεν λαμβάνονται υπόψη. Λόγω αυτού, στο τέλος η πρόβλεψη για κάθε pixel από το HybridSN θα αντιστοιχεί σε μία από τις ground-truth labels. Προκειμένου να δημιουργήσουμε ένα αποτέλεσμα που να είναι οπτικά πανομοιότυπο με αυτό της ground-truth εικόνας, θα θέλαμε να μπορούμε να προβλέπουμε και τις περιοχές που δεν μας ενδιαφέρουν (μαύρα pixels). Για να το πετύχουμε αυτό εισάγαμε στην ανωτέρω function έναν αριθμό από τυχαία σημεία που αντιστοιχούν σε μαύρα pixel. Το αποτέλεσμα είναι να έχουμε μία επιπλέον τάξη ταξινόμησης (άρα 15 συνολικά αντί για 14 που είχαμε αρχικά).

Προκειμένου να καταστεί εφικτή η παρακολούθηση της προόδου μάθησης, τα training accuracy, training loss, validation accuracy και validation loss αποθηκεύονται μετά το τέλος κάθε εποχής. Επίσης, ανάλογα με την validation accuracy, το μοντέλο μας αποθηκεύεται μετά το πέρας κάθε εποχής στην οποία η validation accuracy παρουσιάζει βελτίωση.

Οι εικόνες που αποτελούν το training set έχουν 176 φασματικές ζώνες (οι οποίες γίνονται 30 μετά την εφαρμογή του PCA). Η εικόνα "Dioni" έχει χωρικές διαστάσεις  $250 \times 1376$  και η εικόνα "Loukia"  $249 \times 945$ . Λόγω του ότι στην "Loukia" όλες οι ground-truth labels υπάρχουν ενώ στην "Dioni" 2 λείπουν, αποφασίστηκε να εκπαιδεύσουμε ένα μόνο δίκτυο χρησιμοποιώντας την πρώτη εικόνα. Επιπρόσθετα, οι ground-truth labels χρησιμοποιήθηκαν από το δίκτυο αφού πρώτα

τροποποιήθηκαν ώστε να είναι σε μορφή 1-hot encoding .

Όλες οι άλλες παράμετροι, καθώς και οι μέθοδοι που χρησιμοποιούνται για την αξιολόγηση των αποτελεσμάτων μας και οι οποίες δεν αναφέρονται ρητά, είναι ίδιες με αυτές που παρουσιάζονται στο paper.

### 3 Πειράματα και Αποτελέσματα

Στα πειράματά μας αποφασίσαμε να χρησιμοποιήσουμε  $B=30$  και mini-batch size 64. Όπως αναφέρθηκε στην παράγραφο "Δομή Δικτύου και Σχεδιαστικές Αποφάσεις" κρίθηκε χρήσιμο να εισάγουμε μια νέα τάξη η οποία περιλαμβάνει pixels για τα οποία δεν ενδιαφερόμαστε (μαύρα). Έγιναν αρκετές απόπειρες προκειμένου να βρεθεί ο κατάλληλος αριθμός από τέτοια pixels που χρειάζεται να εισάγουμε στην νέα αυτή τάξη. Αρχικά δοκιμάσαμε να τα χρησιμοποιήσουμε όλα αλλά το πείραμα αυτό απέτυχε τελείως καθώς καταλήξαμε να έχουμε πρόβλημα λόγω class imbalance, που οδήγησε σε σχεδόν μηδενική ακρίβεια. Τιμές μεγαλύτερες των 20000 pixels θα οδηγήσουν σε εξάντληση της μνήμης. Για τον λόγο αυτό, και επειδή βρέθηκε ότι για την εικόνα "Loukia" ο αριθμός των pixels που αντιστοιχούν σε όλες τις 14 τάξεις είναι περίπου 30000, αποφασίστηκε να δοκιμάσουμε να εισάγουμε αρχικά 10000 και στην συνέχεια 2000 μαύρα pixels σε αυτή την νέα τάξη. Επειδή δεν παρατηρήθηκαν ουσιαστικές διαφορές μεταξύ αυτών των δύο περιπτώσεων, αποφασίσαμε να χρησιμοποιήσουμε μόνο 2000 μαύρα pixels λόγω του μειωμένου υπολογιστικού χρόνου που απαιτείται σε αυτή την περίπτωση.

Τα πειράματα που πραγματοποιήσαμε αρχικά έτρεξαν σε ένα υπολογιστή με 4 GB RAM. Για το λόγο αυτό, και προκειμένου να αποκτήσουμε εμπειρία για τις καλύτερες δυνατές παραμέτρους που θα πρέπει να χρησιμοποιήσουμε στα τελευταία πειράματα που θα πραγματοποιηθούν στο google colab, ένα παράθυρο  $5*5$  χρησιμοποιήθηκε αρχικά και η εκπαίδευση του δικτύου έγινε με την εικόνα "Loukia". Η εικόνα αυτή παρουσιάζεται στο Σχ.2.

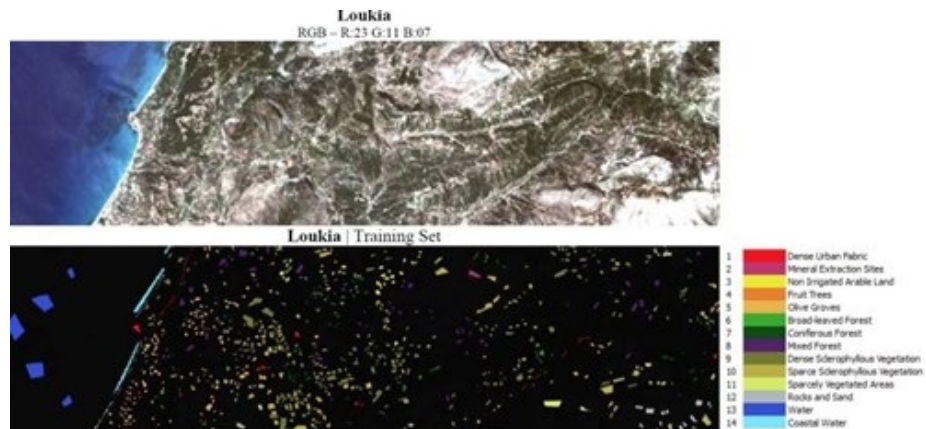


Figure 2: Η εικόνα "Loukia"

Τα πειράματα που πραγματοποιήσαμε εκτός cloudeίναι τα ακόλουθα:

1. Χρησιμοποιήθηκε  $lr=0.001$  το οποίο κατέληξε σε overfitting σε λιγότερο από 20 εποχές.
2. Χρησιμοποιήθηκε  $lr=0.0001$  το οποίο άρχισε να κάνει overfitting μετά από 70-80 εποχές. Στην εποχή 100 το  $lr$  μειώθηκε σε 0.00001 και η εκπαίδευση του δικτύου συνεχίστηκε για άλλες 20 εποχές. Στις εποχές 100 έως 120 δεν σημειώθηκε καμία ουσιαστική αλλαγή και είχαμε stagnation. Τέλος, προκειμένου να επιβεβαιώσουμε ότι περαιτέρω μείωση του  $lr$  δεν οδηγεί σε καλύτερη ακρίβεια δοκιμάσαμε να μειώσουμε το  $lr$  σε 0.000001 και επιχειρήσαμε να εκπαιδεύσουμε το δίκτυο για ακόμα 20 εποχές. Τις εποχές 120-140 δεν παρατηρήθηκε καμία αλλαγή, όπως και πριν.
3. Χρησιμοποιήθηκε  $lr=0.005$  το οποίο είναι μεταξύ των δύο παραπάνω περιπτώσεων. Το αποτέλεσμα ήταν και πάλι overfitting σε λιγότερο από 20 εποχές.
4. Χρησιμοποιήθηκε  $lr=0.00001$  το οποίο σημείωσε αργή πρόοδο για τις πρώτες 100 εποχές αφήνοντας αρκετά περιθώρια για περαιτέρω εκπαίδευση. Φτάνοντας στην εποχή 300 και παρατηρώντας την σχεδόν αμελητέα βελτίωση της ακρίβειας αποφασίστηκε να μειωθεί ακόμα περισσότερο η τιμή του  $lr$  σε 0.000001 και μετά από ακόμα 50 εποχές παρατηρήθηκε και πάλι stagnation.
5. Χρησιμοποιήθηκε  $lr=0.00005$  το οποίο παρουσίασε θετικά δείγματα, αρχίζοντας να κάνει overfitting σχεδόν στο τέλος των 100 εποχών. Μειώνοντάς την τιμή του σε 0.00001 και εκπαιδεύοντας το δίκτυο για άλλες 50 εποχές παρατηρούμε ότι και πάλι οδηγούμαστε σε stagnation.
6. Για  $lr=0.0001$  δοκιμάστηκε επίσης η αντικατάσταση όλων των ReLU με ELU αλλά παρουσιάστηκαν και πάλι παρόμοια αποτελέσματα με την δεύτερη περίπτωση, οδηγώντας μας σε overfitting μετά από 30 περίπου εποχές αντί για 70-80.

Αρκετά χρήσιμα συμπεράσματα μπορούν να εξαχθούν από τα παραπάνω πειράματα. Πρώτα από όλα, φαίνεται ότι ένα  $lr=0.00005$  είναι το καλύτερο δυνατό για τα δεδομένα μας. Επίσης, αξίζει να δοκιμαστεί να αντικαταστήσουμε τις ReLU με ELU για αυτό το  $lr$  και να συγκρίνουμε τα αποτελέσματα που προκύπτουν. Η αρχιτεκτονική του δικτύου φαίνεται κατάλληλη στην υπάρχουσα μορφή της αφού μπορούμε να επιτύχουμε τα προαναφερθέντα αποτελέσματα.

Λαμβάνοντας υπόψη τα παραπάνω συμπεράσματα τα επόμενα πειράματα πραγματοποιήθηκαν στο google colab με ένα παράθυρο  $13*13$ . Για το  $lr$  η τιμή 0.0005 επιλέχθηκε για όλα τα πειράματα. Τέλος, όπως και στο paper επιλέχθηκαν να χρησιμοποιηθούν ως metrics της απόδοσης του δικτύου τα: Overall Accuracy (OA), Average Accuracy (AA) και Kappa Accuracy (KA). Πληροφορίες για το τι δείχνει το κάθε metric παρουσιάζονται στο [1]. Τα metrics αυτά αναφέρονται στην απόδοση του αλγόριθμού μας στο test set.

Τα πειράματα που πραγματοποιήθηκαν στο cloud είναι τα ακόλουθα:

1. Επιλέχθηκε να δοκιμαστεί η τιμή 0.3 για το test ratio και με 2000 μαύρα pixels . Η OA που επιτεύχθηκε για το test set ήταν 95.53%, η AA 95.58% και η KA 94.82%. Τις εποχές 50-100 αρχίζει να παρατηρείται stagnation στην validation accuracy και overfitting στην validation loss. Δοκιμάστηκε στην συνέχεια να συνεχιστεί η εκπαίδευση για άλλες 50 εποχές με  $lr=0.000001$  αλλά δεν παρατηρήθηκε καμία βελτίωση.
2. Επιλέχθηκε να δοκιμαστεί η τιμή 0.7 για το test ratio, με 10000 μαύρα pixels. Η OA που επιτεύχθηκε για το test set ήταν 91.13%, η AA 87.24% και η KA 88.54%. Από την εποχή 60 και μετά αρχίζει να παρατηρείται overfitting.
3. Επιλέχθηκε να δοκιμαστεί η τιμή 0.7 για το test ratio, με 2000 μαύρα pixels. Δυστυχώς δεν διατηρήθηκε το αρχείο με τα αποτελέσματα αλλά παρατηρήθηκε ότι όλα τα metrics δεν ήταν τόσο καλά όσο στην περίπτωση με το test ratio=0.3
4. Για test ratio=0.3 και επιλέγοντας να αγνοήσουμε την τάξη με τα μαύρα pixels (άρα έχοντας συνολικά 14 τάξεις χωρίς μαύρα pixels) η OA που επιτεύχθηκε ήταν 97-98%, η καλύτερη που έχει επιτευχθεί μέχρι στιγμής.
5. Για test ratio=0.3 και επιλέγοντας να αγνοήσουμε την τάξη με τα μαύρα pixels και χρησιμοποιώντας ELUs αντί για ReLUs η OA μειώθηκε κατά περίπου 1%.

Τα καλύτερα αποτελέσματα που επιτεύχθηκαν από τα παραπάνω πειράματα παρατηρούνται όταν αγνοήσουμε τα μαύρα pixels.

Στην Εικ.3 παρουσιάζεται το ground truth της εικόνας "Loukia" που χρησιμοποιήθηκε κατά την εκπαίδευση του δικτύου. Σε αυτό, έχουν επιλεγεί μόνο οι περιοχές που ανήκουν σε 1 από τις 14 τάξεις και όλα τα υπόλοιπα pixels έχουν αγνοηθεί (είναι μαύρα). Επίσης, αν επιλέγαμε να κρατήσουμε και τα μαύρα pixels από τις προβλέψεις του δικτύου μας (όπως στο πείραμα 1 προηγουμένως) το αποτέλεσμα θα έμοιαζε όπως αυτό στην Εικ.4. Είναι φανερό ότι σε αυτή την περίπτωση ο αλγόριθμος αποτυγχάνει να αναγνωρίσει σωστά πολλές περιοχές με μαύρα pixels τις οποίες μάλιστα προβλέπει να έχουν το χρώμα του pixel της κοντινότερης τάξης.

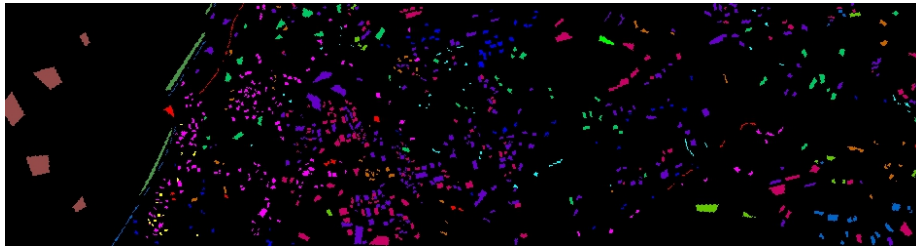


Figure 3: Ground truth για την εικόνα "Loukia"

Επιπρόσθετα, για το καλύτερο αποτέλεσμα, το οποίο είναι αυτό του πειράματος 4, η εικόνα που δίνει ο αλγόριθμος ως πρόβλεψη είναι η Εικ.5. Αξίζει να σημειωθεί

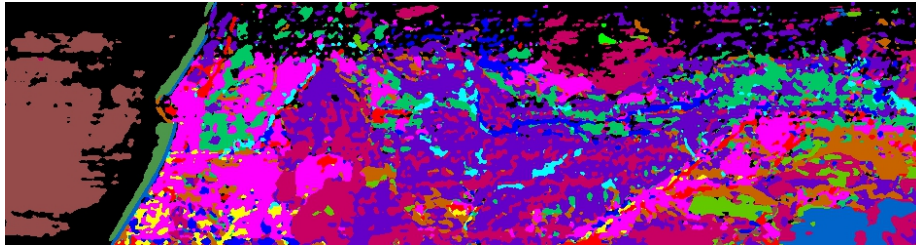


Figure 4: Προβλέψεις για την εικόνα "Loukia" με μαύρα pixels

ότι αν δεν λάβουμε υπόψη τα pixels της εικόνας αυτής που αντιστοιχούν σε μαύρα pixel στην εικόνα 3 (εφαρμόζοντας πχ. μια μάσκα σε αυτά), το αποτέλεσμα που θα πάρουμε είναι όμοιο οπτικά με την Εικ.3, γεγονός που αναμένεται λόγω της μεγάλης ακρίβειας των προβλέψεων αυτής της μεθόδου. Η περίπτωση αυτή παρουσιάζεται στην Εικ.6. Αντίστοιχα, ακόμα και αν λαμβάναμε υπόψη την εικόνα 4 με τα μαύρα pixels, αν εφαρμόζαμε μια μάσκα όπως προηγουμένως, το αποτέλεσμα θα ήταν και πάλι οπτικά πανομοιότυπο καθώς και σε αυτή την περίπτωση έχουμε αρκετά μεγάλη ακρίβεια (η μείωση στην ακρίβεια αυτής της προσέγγισης οφείλεται κυρίως στην αδυναμία του αλγόριθμου να προβλέψει σωστά την τοποθεσία των μαύρων pixels). Η επιλογή για το εάν θα οπτικοποιήσουμε ή όχι τα μαύρα pixels είναι ζήτημα προτίμησης και αφήνεται στον αναγνώστη.

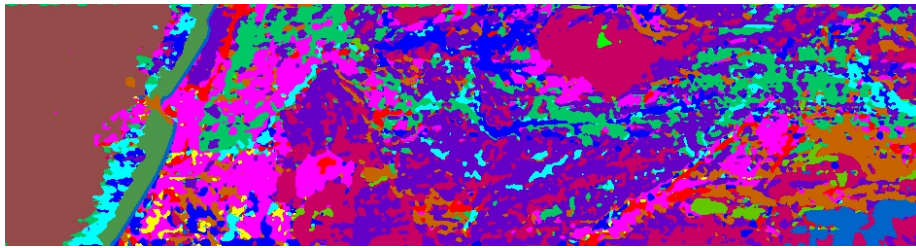


Figure 5: Προβλέψεις για την εικόνα "Loukia" χωρίς μαύρα pixels

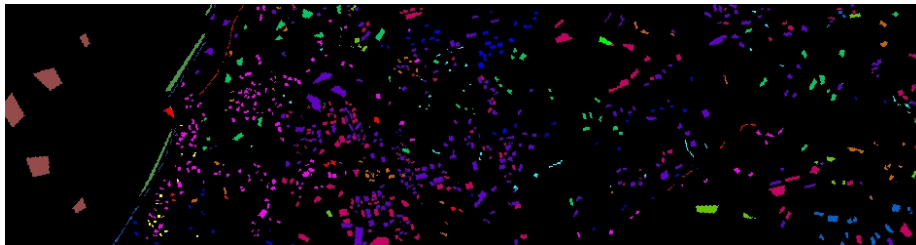


Figure 6: Προβλέψεις για την εικόνα "Loukia" μη λαμβάνοντας υπόψη τα μαύρα pixelτων ground-truth labels



Τέλος, ένα γράφημα με το πώς μεταβλήθηκαν κατά την διάρκεια της εκπαίδευσης του δικτύου οι training και validation accuracy καθώς επίσης και οι training και validation loss, παρουσιάζονται στα Σχ.7 και 8 αντίστοιχα.

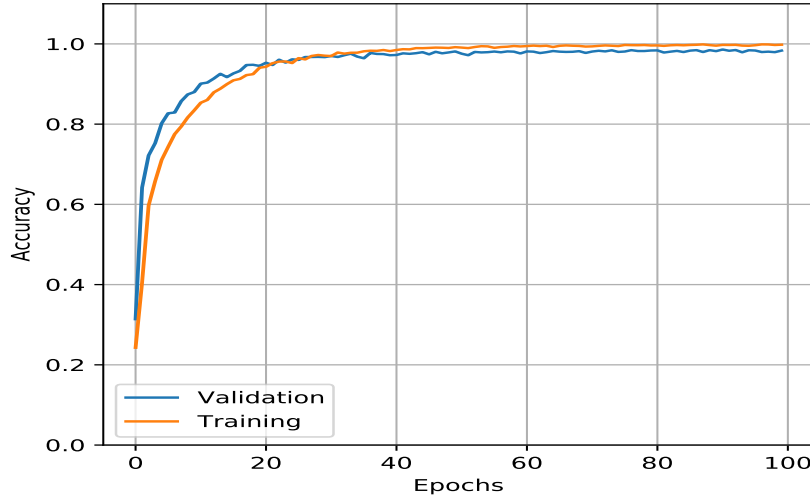


Figure 7: Γράφημα των validation και training accuracy κατά την διάρκεια της εκπαίδευσης του δικτύου

Συνοπτικά, τα καλύτερα αποτελέσματα που επιτύχαμε με την παραπάνω μέθοδο είναι 97.85% OA, 96.58% AA και 97.45% KA. Πληροφορίες για το πόσα παραδείγματα από την κάθε τάξη ταξινομήθηκαν σωστά παρέχονται στο classification report.

Επίσης, αξίζει να σημειωθεί ότι δοκιμάστηκε και το ίδιο πείραμα, αλλάζοντας μόνο το μέγεθος του παραθύρου σε  $11 \times 11$ . Διαπιστώθηκε ότι η ακρίβεια που επιτυγχάνουμε (95%) βρίσκεται μεταξύ των πειραμάτων με μέγεθος παραθύρων  $5 \times 5$  και  $13 \times 13$  (92% και 98% αντίστοιχα), δεδομένου ότι όλες οι άλλες παράμετροι είναι ίδιες και στα 3 αυτά πειράματα. Είναι πιθανό η ακρίβεια που επιτύχαμε να μπορεί να βελτιωθεί περαιτέρω με την χρήση ενός μεγαλύτερου παραθύρου όπως πχ.  $25 \times 25$ , που συνίσταται στο [1]. Χρειαζόμαστε περισσότερη μνήμη RAM προκειμένου να επαληθεύσουμε τον ισχυρισμό αυτό. Τέλος πρέπει να επισημάνουμε ότι δεν κατέστη εφικτό να εξάγουμε συμπεράσματα για το πόσο καλές προβλέψεις κάνει το δίκτυό μας στην δεύτερη εικόνα του training set καθώς αντιμετωπίσαμε και πάλι πρόβλημα εξάντλησης της μνήμης.

Τέλος, οι προβλέψεις του δικτύου στις εικόνες ελέγχου παρουσιάζονται στις Εικ.9, 10 και 11. Αν θέλαμε οι εικόνες αυτές να περιλαμβάνουν και μαύρα pixels τότε θα πρέπει να επιτρέψουμε κατά την εκπαίδευση του δικτύου την παρουσία αυτής της επιπλέον τάξης, όπως αναφέρθηκε προηγουμένως. Το αποτέλεσμα θα

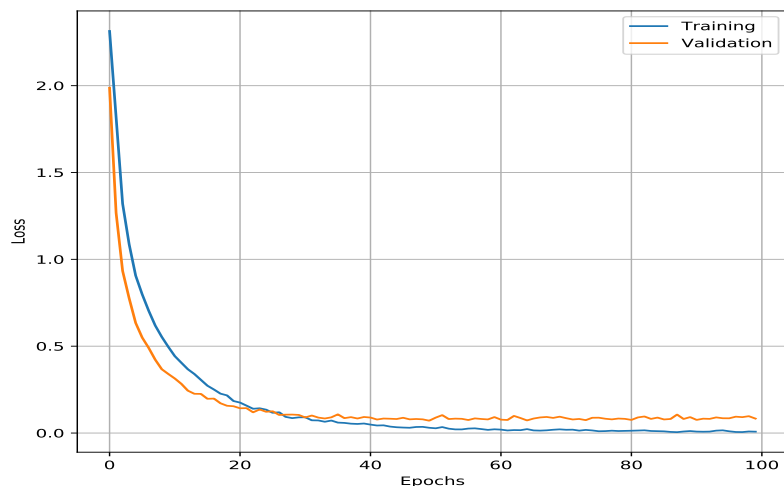


Figure 8: Γράφημα των validation και training loss κατά την διάρκεια της εκπαίδευσης του δικτύου

είναι παρόμοια οπτικά με την Εικ.4. Δεδομένης της μεγάλης ακρίβειας που έχει το δίκτυό μας στο test set αναμένεται ότι θα μπορεί να κάνει επιτυχημένα generalize και επομένως, τα αποτελέσματα στις εικόνες ελέγχου μπορούν να θεωρηθούν αξιόπιστα. Επίσης, δύναται και πάλι να χρησιμοποιήσουμε μια μάσκα προκειμένου να λάβουμε υπόψη μόνο συγκεκριμένες περιοχές ενδιαφέροντος. Όλα τα αρχεία έχουν ανέβει στο google drive. Ο σύνδεσμος όπου μπορούν να βρεθούν είναι [https://drive.google.com/drive/folders/1VKKUbNgGqyMCRkxgB\\_Ncr\\_MUDzc\\_rLtG?usp=sharing](https://drive.google.com/drive/folders/1VKKUbNgGqyMCRkxgB_Ncr_MUDzc_rLtG?usp=sharing).



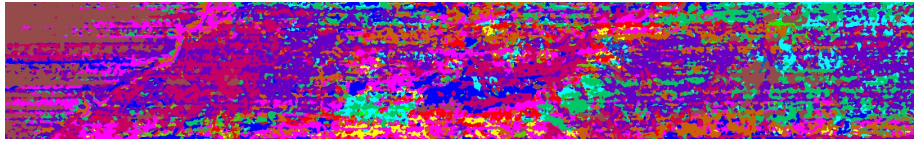


Figure 9: Προβλέψεις του δικτύου για την εικόνα "Erato"

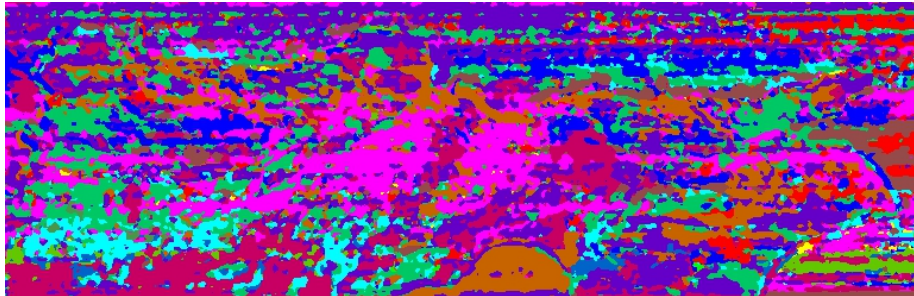


Figure 10: Προβλέψεις του δικτύου για την εικόνα "Nefeli"

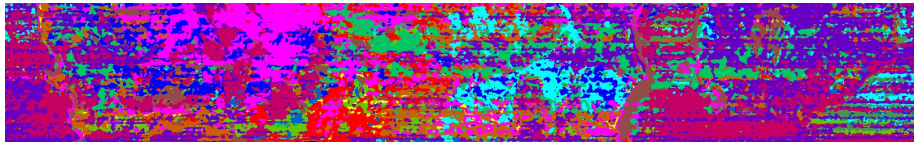


Figure 11: Προβλέψεις του δικτύου για την εικόνα "Kirki"

## References

- [1] S. K. Roy, G. Krishna, S. R. Dubey and B. B. Chaudhuri, "HybridSN: Exploring 3-D-2-D CNN Feature Hierarchy for Hyperspectral Image Classification," in IEEE Geoscience and Remote Sensing Letters, vol. 17, no. 2, pp. 277-281, Feb. 2020, doi: 10.1109/LGRS.2019.2918719.