ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Αναγνώριση Προτύπων

3η Εργαστηριακή Άσκηση

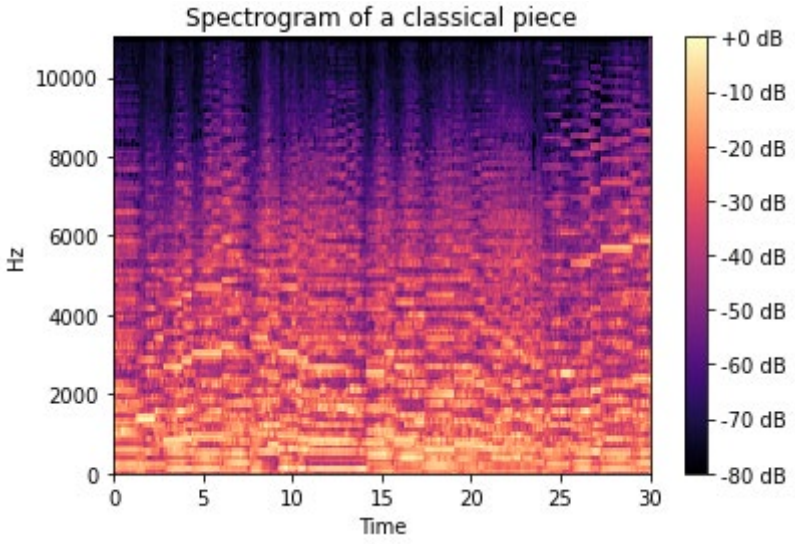
Ονοματεπώνυμο: Ευάγγελος Τσόγκας

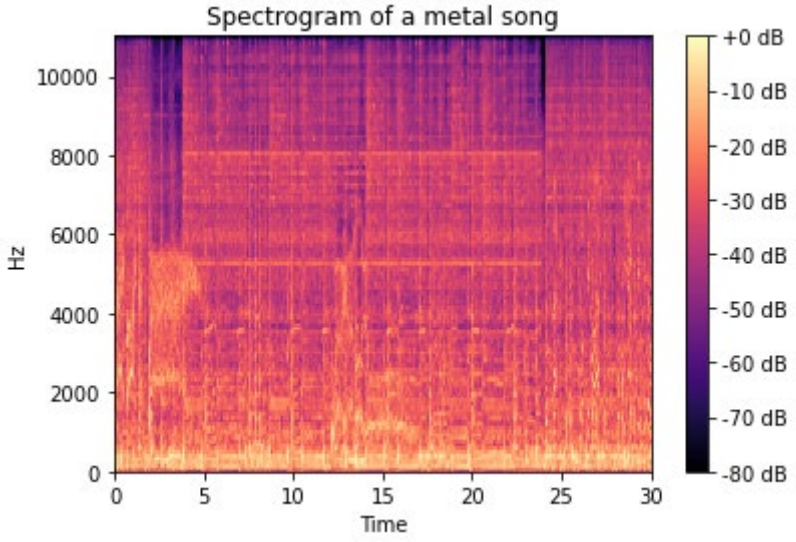
Αριθμός Μητρώου: 03400120

Φασματογραφήματα στην κλίμακα mel

Τα φασματογραφήματα οπτικοποιούν το φάσμα συχνοτήτων ενός σήματος στο πέρασμα του χρόνου. Ο άξονας y δίνει πληροφορία για τη συχνότητα και επίσης μέσω χρωματικής διαβάθμισης δίνεται πληροφορία και για την ένταση του ήχου, όπου σε πιο ανοιχτόχρωμα σημεία αντιστοιχεί μεγαλύτερη ένταση, ενώ σε σκουρόχρωμα μικρότερη.

Στις παρακάτω εικόνες φαίνονται τα mel spectrograms για ένα κομμάτι κλασικής μουσικής και για ένα metal, αντίστοιχα. Μια ενδιαφέρουσα διαφορά είναι πως στο κομμάτι metal παρατηρούμε κάθετες γραμμές που δηλώνουν ξαφνικές αυξήσεις στην ένταση του ήχου. Κάτι τέτοιο είναι λογικό, αφού η metal μουσική χαρακτηρίζεται από έντονους ρυθμούς. Αντίθετα, στην κλασική μουσική υπάρχει μια πιο συνεχής ροή του ήχου οπότε βλέπουμε μικρά οριζόντια τμήματα στα οποία η ένταση παραμένει σταθερή ή έχει μικρές μεταβολές. Μια άλλη εμφανής διαφορά είναι ότι στο κομμάτι κλασικής μουσικής είναι χαμηλότερη η ένταση σε υψηλές συχνότητες.

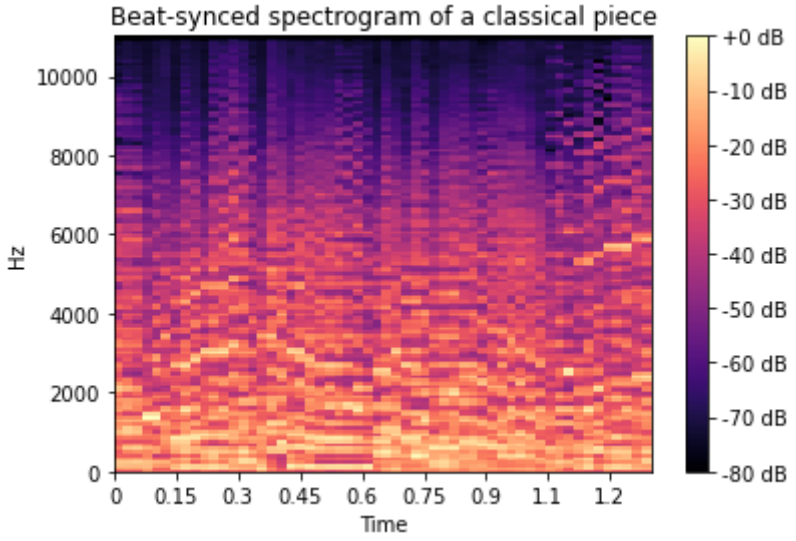


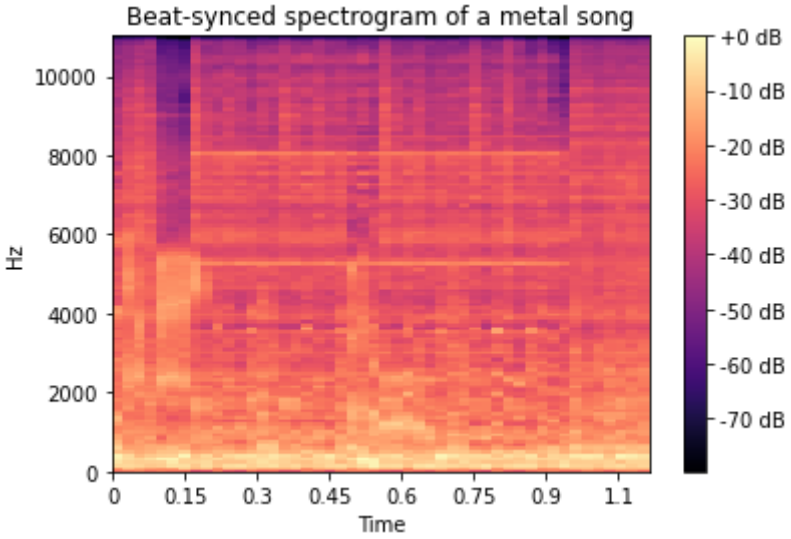


Συγχρονισμός φασματογραφημάτων στο ρυθμό μουσικής

Τα φασματογραφήματα που είδαμε έχουν διάσταση 128 x 1293, δηλαδή έχουν 1293 χρονικά βήματα. Δεν είναι αποδοτικό να εκπαιδεύσουμε ένα LSTM σε αυτά τα δεδομένα, καθώς τα χρονικά βήματα είναι πάρα πολλά και το dataset πολύ μικρό συγκριτικά. Επομένως, το δίκτυο δεν θα είναι ικανό να μάθει τόσες πολλές παραμέτρους με τόσα λίγα δεδομένα.

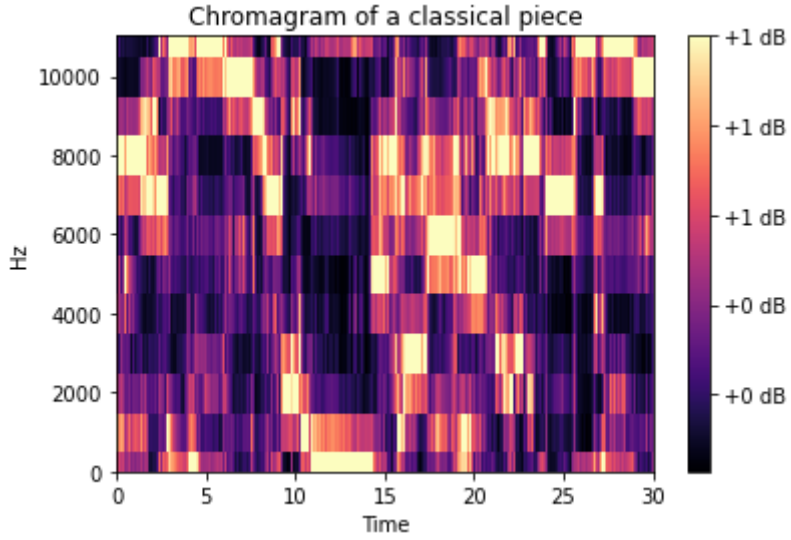
Στις παρακάτω εικόνες φαίνονται τα αντίστοιχα beat-synced spectrograms για τα προηγούμενα κομμάτια μουσικής, όπου έχουμε μειώσει τα χρονικά βήματα παίρνοντας τη διάμεσο ανάμεσα στα σημεία που χτυπάει ο ρυθμός. Τα χρονικά βήματα σε αυτή την περίπτωση είναι 56 για το κλασικό κομμάτι και 48 για το metal. Η γενική μορφή των φασματογραφημάτων δεν έχει αλλάξει, αλλά είναι εμφανές πως έχει χαθεί πληροφορία, αφού οι εικόνες φαίνονται θολές σε σύγκριση με τις αρχικές.

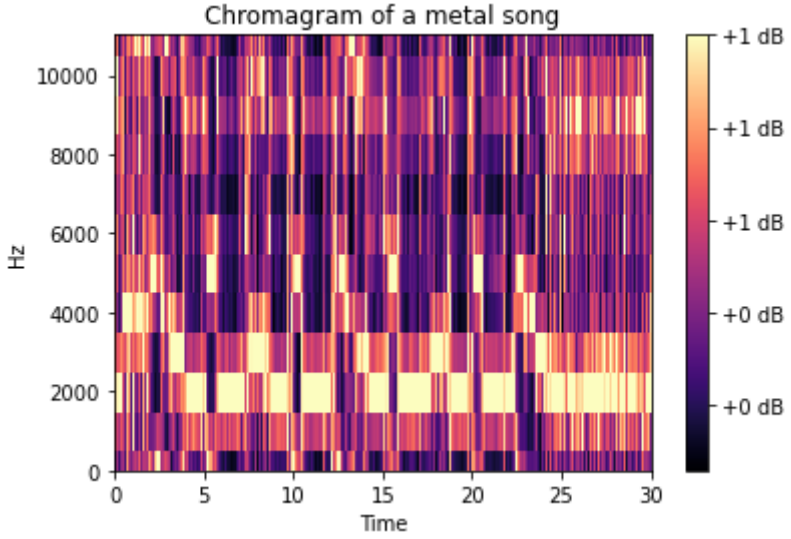




Χρωμογραφήματα

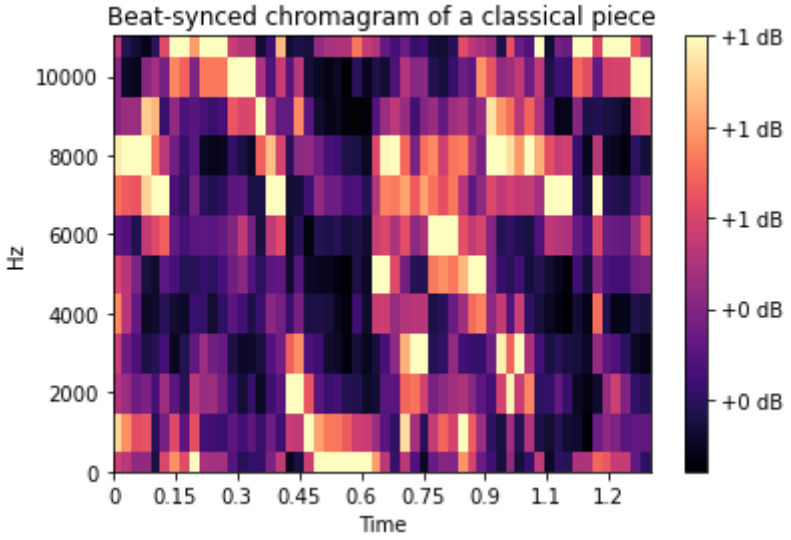
Στις παρακάτω εικόνες φαίνονται τα χρωμογραφήματα για τα δύο κομμάτια μουσικής.

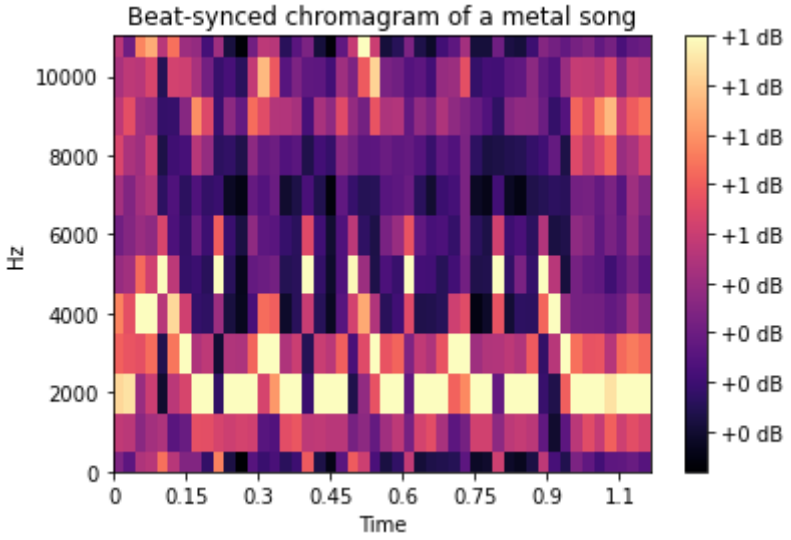




Από τα χρωμογραφήματα μπορούμε να παρατηρήσουμε πληροφορίες σχετικά με 12 διαφορετικές νότες. Για την ακρίβεια, ένας άνθρωπος αντιλαμβάνεται δύο νότες με παρόμοιο χρώμα αν αυτές διαφέρουν κατά μια οκτάβα. Παρατηρούμε πως στο κλασικό κομμάτι υπάρχει αυτή η ομοιότητα σε αρκετές περιοχές κατά το πέρασμα του χρόνου, σε αντίθεση με το metal κομμάτι όπου παρατηρούνται συχνές αλλαγές. Με μια απλή παρατήρηση των εικόνων φαίνεται πως τα χρωμογραφήματα μάλλον βοηθούν αρκετά στη διάκριση των δύο κομματιών.

Επίσης, παρακάτω φαίνονται και τα χρωμογραφήματα για τα beat-synced δεδομένα, όπου όπως και πριν παρατηρούμε μειωμένη πληροφορία, αλλά χωρίς να αλλοιώνεται σημαντικά η αρχική.





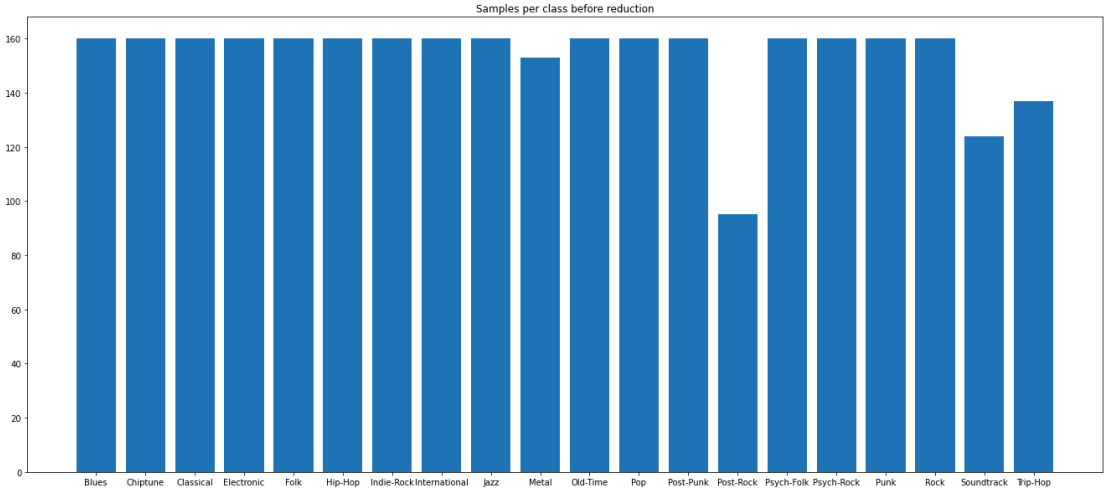
Φόρτωση και ανάλυση δεδομένων

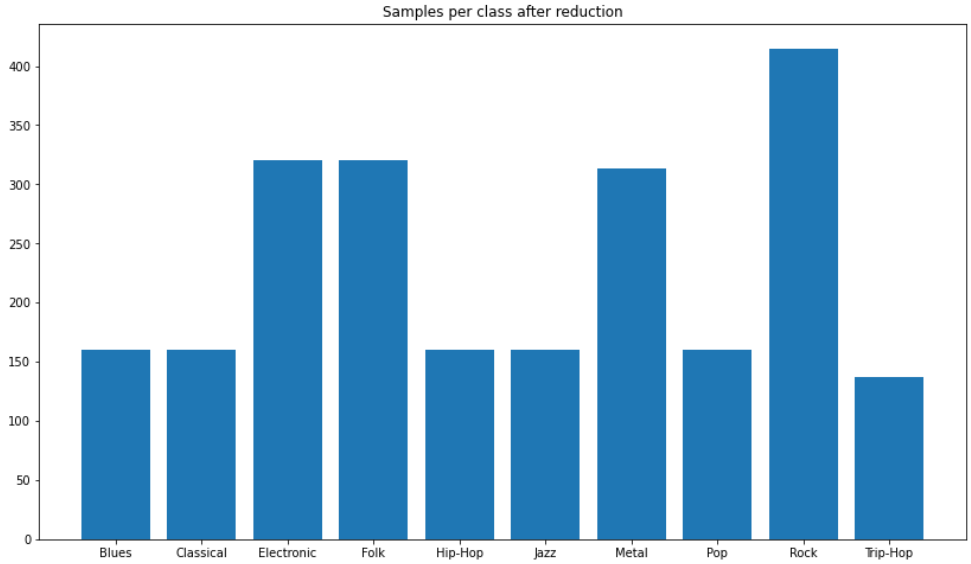
Κατά τη φόρτωση των δεδομένων τα χωρίζουμε σε train και validation. Αυτό γίνεται διαλέγονται τον αριθμό που αντιστοιχεί στο 20% του μεγέθους του dataset και χρησιμοποιώντας αυτόν τον αριθμό σαν split point, κρατάμε τα δεδομένα μέχρι το split point για validation, δηλαδή το 20% των συνολικών δεδομένων και τα υπόλοιπα τα κρατάμε για training. Τα δεδομένα τα κάνουμε shuffle πριν γίνει το split, ώστε να είναι τυχαία η κατανομή των κατηγοριών στα δύο datasets για σωστή εκπαίδευση. Παρ' όλα αυτά, κατά το debugging μπορούμε να θέτουμε ένα random seed για επαναληψιμότητα των αποτελεσμάτων.

Στο PyTorch Dataset κάνουμε zero padding των δεδομένων. Ο λόγος είναι ότι οι ακολουθίες είναι διαφορετικού μεγέθους. Όπως είδαμε πριν για παράδειγμα το κομμάτι κλασικής μουσικής που επιλέξαμε έχει 56 χρονικά βήματα, ενώ το metal 48. Για την εκπαίδευση των μοντέλων, όμως, θέλουμε όλες οι ακολουθίες να έχουν το ίδιο μέγεθος. Τα δείγματα που επιστρέφει ο κώδικας, ανεξαρτήτως dataset που θα χρησιμοποιήσουμε, έχουν τη μορφή (batch size x time steps x frequencies), όπου το batch size είναι 32.

Κλάσεις οι οποίες μοιάζουν μεταξύ τους συγχωνεύονται καθώς το νευρωνικό δίκτυο δεν θα είναι σε θέση να τις ξεχωρίσει, όπως για παράδειγμα παραλλαγές τις Rock μουσικής. Επίσης, κλάσεις με πολύ λίγα δείγματα αφαιρούνται καθώς θα οδηγήσουν σε imbalanced dataset και πάλι το νευρωνικό δίκτυο δεν θα μπορεί να τις αναγνωρίσει. Μεγάλο ρόλο φυσικά παίζει το γεγονός ότι τα δεδομένα γενικά είναι λίγα. Αν είχαμε περισσότερα δεδομένα πολύ πιθανόν να μπορούσε να ανταπεξέλθει ικανοποιητικά το δίκτυο, ακόμα και σε αυτές τις περιπτώσεις.

Στα παρακάτω γραφήματα φαίνεται ο αριθμός των δειγμάτων ανά κλάση πριν και μετά τη συγχώνευση/διαγραφή, αντίστοιχα.





Αναγνώριση μουσικού είδους με Long Short-Term Memory (LSTM) Network

Χρησιμοποιώντας τα δεδομένα που περιεγράφηκαν εκπαιδεύω 4 μοντέλα LSTM με τα εξής sets δεδομένων, αντίστοιχα:

a) Mel spectrograms
b) Beat-synced mel spectrograms
c) Beat-synced chromagrams
d) Fused beat-synced mel spectrograms + chromagrams

Λεπτομέρειες για τις παραμέτρους και την αρχιτεκτονική του μοντέλου που χρησιμοποιήθηκε και στις 4 περιπτώσεις φαίνονται στον παρακάτω πίνακα:

Παράμετρος	Τιμή
Bidirectional	True
LSTM layers	2
LSTM size	128
Dropout	0.2
L2 regularization	0.001
Early stopping patience	5
Optimizer	Adam
Learning rate	0.0001
Batch size	32

Στα παρακάτω screenshots φαίνονται τα αποτελέσματα του classification για τα 4 μοντέλα. Το καλύτερο μοντέλο με βάση τις μετρικές accuracy, macro-averaged f1 και weighted-averaged f1 είναι αυτό το οποίο εκπαιδεύτηκε με το συνδυασμό των beat-synced φασματογραφημάτων και χρωμογραφημάτων (Εικόνα 4).

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.45	0.25	0.32	40
2	0.26	0.62	0.37	80
3	0.27	0.68	0.39	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.42	0.58	0.49	78
7	0.00	0.00	0.00	40
8	0.37	0.20	0.26	103
9	0.00	0.00	0.00	34
accuracy			0.31	575
macro avg	0.18	0.23	0.18	575
weighted avg	0.23	0.31	0.24	575

Εικόνα 1: Αποτελέσματα χρησιμοποιώντας τα φασματογραφήματα.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.52	0.38	0.43	40
2	0.32	0.74	0.45	80
3	0.35	0.64	0.45	80
4	0.67	0.05	0.09	40
5	0.00	0.00	0.00	40
6	0.39	0.68	0.49	78
7	0.00	0.00	0.00	40
8	0.35	0.27	0.31	103
9	0.00	0.00	0.00	34
accuracy			0.36	575
macro avg	0.26	0.28	0.22	575
weighted avg	0.29	0.36	0.28	575

Εικόνα 2: Αποτελέσματα χρησιμοποιώντας τα beat-synced φασματογραφήματα.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.00	0.00	0.00	40
2	0.00	0.00	0.00	80
3	0.21	0.65	0.32	80
4	0.00	0.00	0.00	40
5	0.00	0.00	0.00	40
6	0.27	0.56	0.36	78
7	0.00	0.00	0.00	40
8	0.16	0.25	0.20	103
9	0.00	0.00	0.00	34
accuracy			0.21	575
macro avg	0.06	0.15	0.09	575
weighted avg	0.09	0.21	0.13	575

Εικόνα 3: Αποτελέσματα χρησιμοποιώντας τα beat-synced χρωμογραφήματα.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	40
1	0.44	0.53	0.48	40
2	0.40	0.74	0.52	80
3	0.32	0.53	0.40	80
4	0.24	0.35	0.29	40
5	0.67	0.05	0.09	40
6	0.53	0.59	0.56	78
7	0.00	0.00	0.00	40
8	0.38	0.38	0.38	103
9	0.00	0.00	0.00	34
accuracy			0.39	575
macro avg	0.30	0.32	0.27	575
weighted avg	0.33	0.39	0.33	575

Εικόνα 4: Αποτελέσματα χρησιμοποιώντας συνδυασμό των beat-synced φασματογραφημάτων και χρωμογραφημάτων.

Οι μετρικές που φαίνονται στα αποτελέσματα ερμηνεύονται ως εξής:

- Accuracy: Είναι το ποσοστό των σωστά κατηγοριοποιημένων δειγμάτων.
- Precision: Είναι ο λόγος των σωστά κατηγοριοποιημένων δειγμάτων σε μια κλάση προς το συνολικό αριθμό δειγμάτων που έχουν κατηγοριοποιηθεί σε αυτή την κλάση.
- Recall: Είναι ο λόγος των σωστά κατηγοριοποιημένων δειγμάτων σε μια κλάση προς το σύνολο των δειγμάτων που πραγματικά ανήκουν σε αυτή την κλάση.

- f1-score: Είναι ο αρμονικός μέσος των precision και recall και χρησιμοποιείται γιατί γενικά παρατηρείται trade-off μεταξύ των δύο προηγούμενων.
- Οι micro-averaged μετρικές είναι ο μέσος όρος των συνολικών true positives, false negatives και false positives και έχουν νόημα για παράδειγμα σε multi-label classification. Στη δική μας περίπτωση ουσιαστικά αντιστοιχούν στο accuracy.
- Οι macro-averaged μετρικές είναι ο μέσος όρος των μετρικών για όλες τις κλάσεις, χωρίς να λαμβάνεται υπόψη το imbalance των κλάσεων.
- Οι weighted-averaged μετρικές υπολογίζονται όπως οι macro, αλλά έχουν ως βάρη τον αριθμό των δειγμάτων για κάθε κλάση, ώστε να υπολογίζεται και ο παράγοντας των imbalanced κλάσεων.

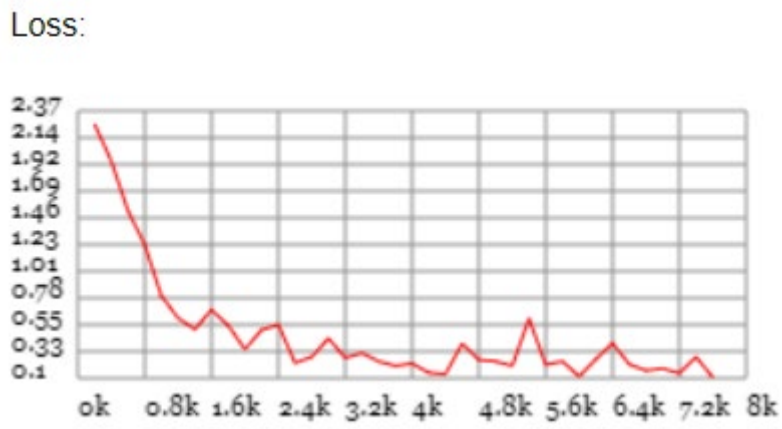
Σε περίπτωση imbalanced dataset το accuracy μπορεί να έχει μεγάλη απόκλιση με το f1-score. Για παράδειγμα, σε ένα πρόβλημα binary classification με ένα δείγμα στη μια κλάση και 99 στην άλλη, αν απλά προβλέπαμε μόνο τη $2^η$ τότε θα είχαμε accuracy 99%, ενώ το f1-score θα ήταν πολύ μικρότερο. Κατά αντιστοιχία και για το micro/macro σε περίπτωση multiclass-classification.

Υπάρχουν βέβαια προβλήματα που μπορεί να μας ενδιαφέρει περισσότερο το recall ή το precision για κάθε κλάση ξεχωριστά και όχι το accuracy ή το f1 που δίνουν πιο γενική ιδέα των αποτελεσμάτων. Για παράδειγμα, υψηλότερο recall είναι σίγουρα πιο σημαντικό σε περίπτωση διάγνωσης καρκίνου από εικόνες, καθώς δεν θέλουμε σε καμία περίπτωση να έχουμε false negatives, αφού αυτό σημαίνει πως ο ασθενής δεν θα θεραπευτεί εγκαίρως. Ένα παράδειγμα που το precision είναι πολύ σημαντική μετρική και θέλουμε να το μεγιστοποιήσουμε είναι στο πρόβλημα του email spam detection. Σε αυτή την περίπτωση θέλουμε να ελαχιστοποιήσουμε τα false positives, καθώς δεν θα πρέπει να υπάρξει σημαντικό μήνυμα που θα αναγνωριστεί ως spam.

Στο δικό μας πρόβλημα οι μετρικές accuracy, macro-averaged f1 και weighted-averaged f1 είναι αρκετές για να αξιολογήσουμε τα αποτελέσματα, αν και το accuracy μάλλον κάνει λίγο overestimate καθώς το dataset είναι ελαφρώς imbalanced. Παρ' όλα αυτά, μπορούμε να πάρουμε χρήσιμη πληροφορία και από το f1 score για κάθε κλάση ξεχωριστά, καθώς βλέπουμε πως σε αρκετές κλάσεις με λίγα δεδομένα δεν έχει κατηγοριοποιηθεί κανένα δείγμα. Αυτό το πρόβλημα, βέβαια, οφείλεται και στο γεγονός πως το dataset είναι σχετικά μικρό.

2D CNN

Πριν εκπαιδεύσουμε το δικό μας CNN ρίχνουμε μια ματιά στο ConvNetJS MNIST demo. Στο συγκεκριμένο παράδειγμα εκπαιδεύεται ένα CNN δίκτυο το οποίο αναγνωρίζει ψηφία από το 0 έως το 9. Στην παρακάτω εικόνα φαίνεται ένα στιγμιότυπο από το learning curve.



Το δίκτυο αυτό αποτελείται από δύο convolutional layers τα οποία ακολουθούνται από την ReLu και τη λειτουργία του pooling. Στο τέλος του δικτύου υπάρχει ένα απλό fully connected layer και η softmax ώστε να γίνει τελικά το classification. Τα convolutional layers ουσιαστικά εφαρμόζουν φίλτρα πάνω στην εικόνα ώστε να εξάγουν χρήσιμες αναπαραστάσεις και το pool χρησιμεύει για τη μείωση της διάστασης των χαρακτηριστικών καθώς προχωράμε πιο βαθιά στο δίκτυο. Παρακάτω φαίνεται ένα παράδειγμα του ψηφίου 4 και της αναπαράστασής του εσωτερικά στο δίκτυο.

Στην εικόνα αριστερά φαίνεται ένα δείγμα από το ψηφίο 4 το οποίο δίνεται σαν είσοδος στο δίκτυο. Στη συνέχεια παρατηρούμε τις ενεργοποιήσεις που παράγονται εφαρμόζοντας 8 φίλτρα στην εικόνα στο πρώτο convolutional layer.

Φαίνεται πως αρχικά εξάγονται χαρακτηριστικά όπως γραμμές και ακμές. Στο δεύτερο convolutional layer εφαρμόζουμε 16 φίλτρα και το αποτέλεσμα φαίνεται παρακάτω.

Το ψηφίο τώρα δεν είναι τόσο εύκολα αναγνωρίσιμο. Αυτό δηλώνει πως σε πιο βαθιά επίπεδα παίρνουμε όλο και πιο γενικές αναπαραστάσεις μιας εικόνας.

Στη συνέχεια υλοποιούμε ένα CNN ώστε να κάνουμε αναγνώριση είδους μουσικής χρησιμοποιώντας φασματογραφήματα. Σε κάθε Convolution block χρησιμοποιούμε ένα convolutional layer, batch normalization, ReRu, και max pooling. Το convolution όπως ήδη αναφέραμε εφαρμόζει διάφορα φίλτρα στην εικόνα και εξάγει τα feature maps τα οποία αποτελούν χρήσιμες αναπαραστάσεις της εικόνας εσωτερικά στο δίκτυο. Ο σκοπός του batch normalization είναι ο ίδιος με το γενικό normalization των δεδομένων όταν εκπαιδεύουμε ένα νευρωνικό δίκτυο. Το να φέρνουμε τα δεδομένα σε μια μορφή με μηδενική μέση τιμή και μοναδιαίο variance αποδεδειγμένα κάνει την εκπαίδευση πιο γρήγορη και πιο σταθερή.

Η ReLu όπως και στα απλά νευρωνικά δίκτυα αποτελεί activation function και ουσιαστικά φιλτράρει αρνητικές τιμές. Τέλος, το max pooling κρατάει τις μέγιστες τιμές των feature maps και ελαττώνει τις διαστάσεις τους.

Παρακάτω φαίνονται τα αποτελέσματα του classification χρησιμοποιώντας το CNN και όπως βλέπουμε υπάρχει σημαντική βελτίωση σε σύγκριση με το LSTM.

Classification report:

	precision	recall	f1-score	support
0	0.30	0.20	0.24	40
1	0.46	0.60	0.52	40
2	0.57	0.57	0.57	80
3	0.44	0.62	0.52	80
4	0.52	0.68	0.59	40
5	0.18	0.17	0.18	40
6	0.61	0.58	0.59	78
7	0.18	0.07	0.11	40
8	0.41	0.38	0.39	103
9	0.33	0.26	0.30	34
accuracy			0.45	575
macro avg	0.40	0.41	0.40	575
weighted avg	0.43	0.45	0.43	575

Εικόνα 5: Αποτελέσματα CNN χρησιμοποιώντας τα φασματογραφήματα.

Εκτίμηση συναισθήματος – συμπεριφοράς με παλινδρόμηση

Για την εκτίμηση συναισθήματος εκπαιδεύουμε κάθε μοντέλο τρεις φορές έτσι ώστε να προβλέπει τα valence, energy και danceability, αντίστοιχα. Επίσης, για το LSTM μοντέλο χρησιμοποιήθηκαν τα beat-synced φασματογραφήματα, ενώ για το CNN τα αρχικά. Για την αξιολόγηση χρησιμοποιήθηκε το 20% των δεδομένων κάθε φορά και τα αποτελέσματα με τη μετρική spearman correlation φαίνονται στον παρακάτω πίνακα.

Μοντέλο	Valence	Energy	Danceability	Mean
LSTM	0.252	0.706	0.455	0.471
CNN	**0.561**	**0.711**	**0.668**	**0.646**

Όπως παρατηρούμε το CNN και σε αυτό το task είχε καλύτερες επιδόσεις από το LSTM.

Μεταφορά Γνώσης (Transfer Learning)

Στο paper: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf, εξήγαγαν τα ακόλουθα συμπεράσματα σχετικά με το transfer learning:

- Το transfer learning επηρεάζεται αρνητικά αν επιλέξουμε layers από τη μέση του νευρωνικού δικτύου ή από πολύ υψηλά επίπεδα που είναι πολύ εξειδικευμένα στο task για το οποίο αρχικά εκπαιδεύτηκε το δίκτυο.
- Όσο πιο διαφορετικά είναι τα δύο tasks του transfer learning (ειδικά όταν χρησιμοποιούμε υψηλά, εξιδεικευμένα επίπεδα για το transfer) τόσο πιο αρνητικά επηρεάζεται η επίδοση, αλλά και πάλι είναι καλύτερη από τυχαία αρχικοποίηση βαρών.
- Η αρχικοποίηση με transferred χαρακτηριστικά βελτιώνει την γενίκευση του μοντέλου ακόμα και με fine-tuning για το καινούριο task.

Με βάση τα παραπάνω συμπεράσματα, για να κάνουμε transfer learning στο πρόβλημά μας επιλέγουμε το μοντέλο CNN που εκπαιδεύσαμε στο dataset για αναγνώριση είδους μουσικής με σκοπό να το χρησιμοποιήσουμε για την εκτίμηση του συναισθήματος. Ο λόγος που επιλέγουμε το CNN και όχι το LSTM είναι πως έχει καλύτερα αποτελέσματα στα tasks που θέλουμε να αντιμετωπίσουμε. Ο τρόπος με τον οποίο γίνεται η εκπαίδευση στο νέο dataset είναι κρατώντας τα βάρη των convolutional layers, καθώς δεν θέλουμε να χαθεί η πληροφορία που έχει μάθει στο προηγούμενο dataset, αλλά αντικαθιστώντας τα τελευταία fully connected layers τα οποία είναι εξιδεικευμένα στο προηγούμενο task και κάνουμε fine-tuning στο νέο task για λίγες εποχές. Τα αποτελέσματα με το transfer learning φαίνονται στον παρακάτω πίνακα:

Μοντέλο	Valence	Energy	Danceability	Mean
CNN (Transfer)	**0.594**	**0.790**	**0.681**	**0.688**

Παρατηρούμε πως πράγματι υπάρχει βελτίωση στα αποτελέσματα, ακόμα κι αν το dataset ήταν μικρό.

Multitask Learning

Στο paper: https://arxiv.org/pdf/1706.05137.pdf, εξήγαγαν τα ακόλουθα συμπεράσματα:

- Ένα μοντέλο εκπαιδευμένο για 8 tasks ταυτόχρονα έχει παρόμοια αποτελέσματα με state-of-the-art μοντέλα εκπαιδευμένα για κάθε task ξεχωριστά, και σε μικρά datasets αρκετά καλύτερα από ότι σε μεγάλα.
- Συνδυάζοντας computational blocks από διαφορετικούς τομείς μπορούν να βελτιωθούν τα αποτελέσματα.
- Τα αποτελέσματα βελτιώνονται όταν τα tasks μοιράζονται αρκετές παραμέτρους.

Με βάση τα παραπάνω συμπεράσματα εκπαιδεύουμε ένα CNN για multi-task emotion recognition κρατώντας κοινά τα convolutional layers για τα τρία tasks (πρόβλεψη valence, energy και danceability) και έχοντας διαφορετικά fully connected layers. Επομένως το μοντέλο βγάζει τρία outputs.

Τα αποτελέσματα του multitask learning φαίνονται στον παρακάτω πίνακα:

Μοντέλο	Valence	Energy	Danceability	Mean
CNN (Multitask)	**0.668**	**0.794**	**0.768**	**0.744**

Όπως βλέπουμε υπάρχει βελτίωση σε σχέση με τη μάθηση των task ξεχωριστά, ακόμα και με transfer learning.

Υποβολή στο Kaggle

Το καλύτερο μοντέλο από αυτά που δοκίμασα είναι το CNN που κάνει multitask learning, επομένως επέλεξα αυτό για την υποβολή των προβλέψεων στον διαγωνισμό Kaggle. Το αποτέλεσμα στο leaderboard είναι: **0.71990**

Είναι λίγο χαμηλότερο από το αποτέλεσμα που βγάζω στο validation set που χρησιμοποίησα, αλλά αναμενόμενο, αφού τα δεδομένα στα οποία αξιολογήθηκε είναι τελείως άγνωστα. Το username μου στο Kaggle είναι Evangelos Tsogkas.