Deep Learning and Multimedia Information Analysis

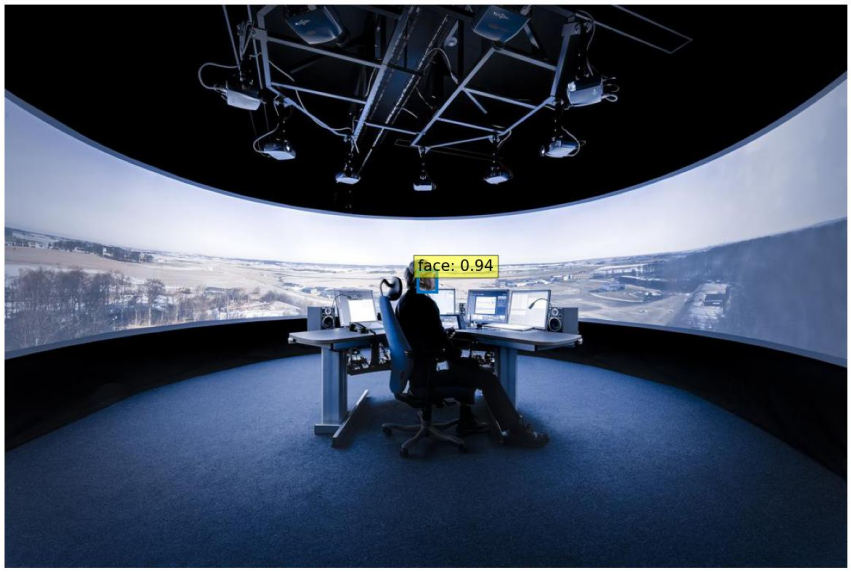
MSc in Artificial Intelligence

Aristotle University of Thessaloniki

Homework 1 & 2

Deep Convolutional Neural Networks & Transformers

DETR : End-to-End Object Detection with Transformers



George Kalitsios

Student Number 62

Thessaloniki 30/5/2021

1. Εισαγωγή

Δεν υλοποιήθηκε εργασία πάνω μόνο σε Βαθιά Συνελικτικά Νευρωνικά Δίκτυα(αν και η αρχική σκέψη ήταν να η υλοποίηση ενος Faster R-CNN ή Mask R-CNN) αλλά συνδυάστηκαν αυτά μαζί με transformers. Για τις πρώτες δυο εργασίες χρησιμοποιήθηκε ο DETR (Detection Transformer) για λυθεί ένα Object Detection task. Η ιδέα ήρθε στην πρώτη παρουσίαση του μαθήματος οπού έγινε αναφορά στον DETR και διαβάζοντας στην συνέχεια το paper και ψάχνοντας παραπάνω την συγκεκριμένη αρχιτεκτονική μου φάνηκε αρκετά ενδιαφέρον να χρησιμοποιηθούν οι transformers που έχουν κυριαρχήσει στο NLP σε ένα παραδοσιακό Computer Vision task όπως είναι το Object Detection. Εκτός από το paper (DETR : End-to-End Object Detection with Transformers) μελετήθηκαν ακόμα τα Attention Is All You Need και BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding για την καλύτερη κατανόηση των transformers και concept όπως Encoder, Decoder, Self Attention, Multi-head attention αλλά και 2 νέα paper που έρχονται να βελτιώσουν επιπλέον την αρχική υλοποίηση του Facebook AI το (Deformable DETR: Deformable Transformers for End-to-End Object Detection) που βελτιώνει την απόδοση στα small objects και το (Efficient DETR: Improving End-to-End Object Detector with Dense Prior) που βγήκε πριν από έναν μηνά (Απρίλιο του 2021) .

2. Dataset

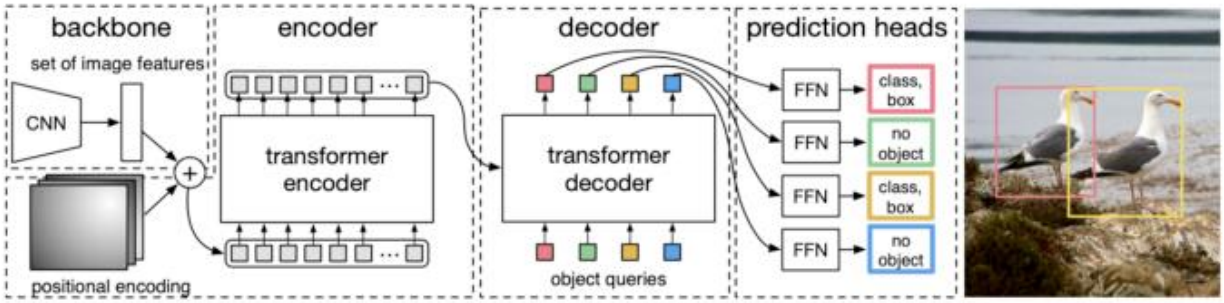
WIDER FACE: A Face Detection Benchmark

Link : http://shuoyang1213.me/WIDERFACE/

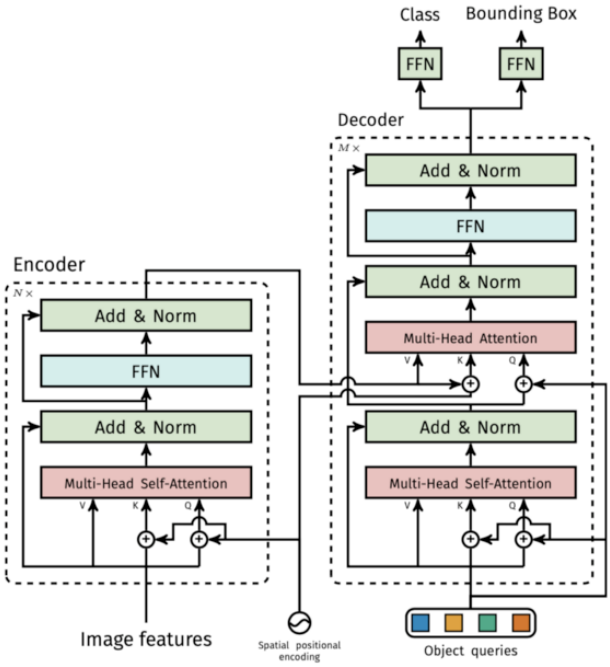
Το dataset έχει 32.203 εικόνες και συνολικά 393.703 πρόσωπα αλλά δεν είναι σε coco format τα annotations. Στον κώδικα που φέρνει το dataset σε coco format κρατήσαμε μόνο εικόνες που έχουν μέχρι 10 πρόσωπα και όχι παραπάνω για να μπορούμε και να ελέγξουμε καλυτέρα τον αριθμό των object queries στην αρχιτεκτονική. Επίσης στα πειράματα με τις παραμέτρους δεν χρησιμοποιήθηκε ολόκληρο το dataset αλλά υποσύνολο αυτού ώστε να μπορεί να γίνει πιο γρήγορα ο πειραματισμός καθώς με ολόκληρο το dataset ο χρόνος ήταν απαγορευτικός.

3. Αρχιτεκτονική του DETR

DETR Architecture



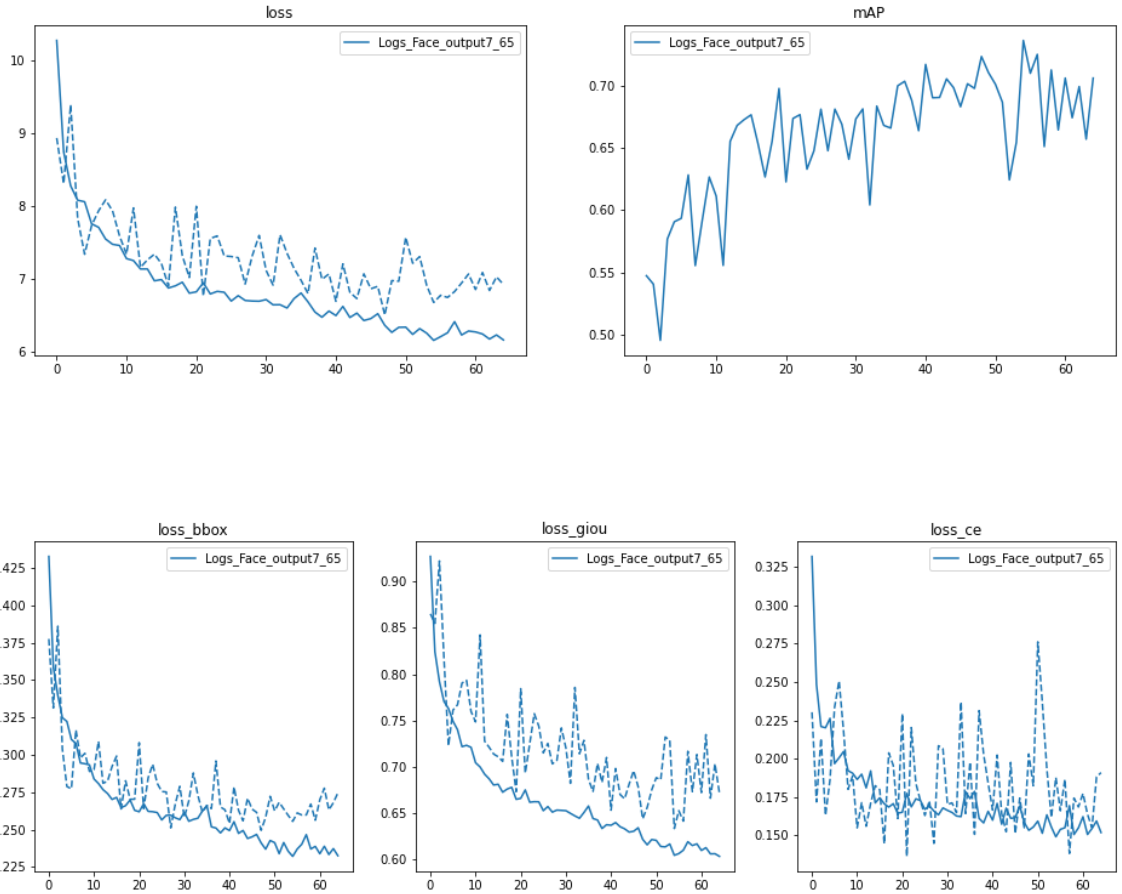
DETR Transformer



4. Πειράματα

- Τα πειράματα εκτελέστηκαν τοπικά σε μια Nvidia RTX 2080 Ti
- Αρχικά δοκιμάστηκε με λίγα δεδομένα να δούμε αν το μοντέλο κάνει overfitting καθώς αναφέρθηκε και στο μάθημα ότι πρέπει να είναι από τα πρώτα που μπορούμε να δοκιμάσουμε όταν κάνουμε debugging.
- **<u>Πείραμα 1ο - Using the entire Dataset:</u>**

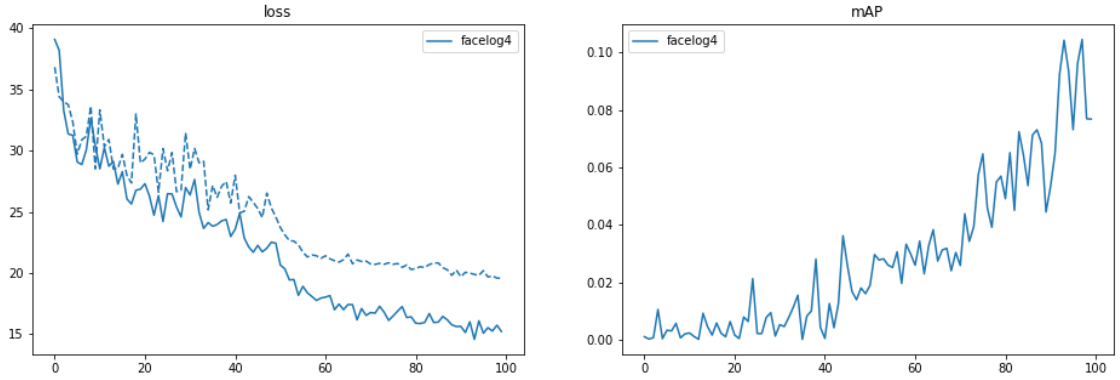
Στο πειραμα αυτό Χρησιμοποιηθηκε ολοκληρο το Dataset ο χρονος εκπαυδευσης για 65 Epochs ηταν 32,5 ωρες καθως για να δεις τις 10000 εικονες του training set χρειαζοταν περιπου 30 λεπτα για 1 epoch. Το Map (Mean Average Precision) αυξανεται καθως προχωραει το training και το loss μειωνεται, παρακατω φαινονται επισης και τα 3 διαφορετικα losses [classification loss, L1 bounding box distance loss, and GIoU (Generalized Intersection over Union) loss].



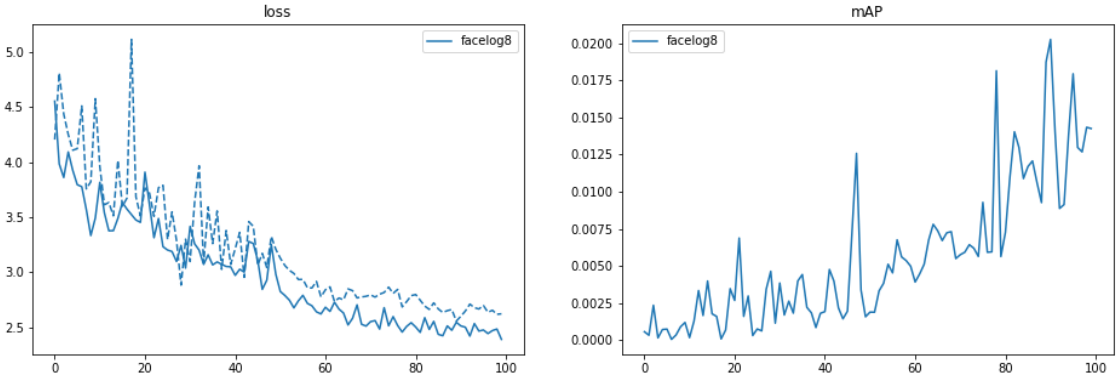
Επειδη το Dataset ηταν αρκετα μεγαλο και ο χρονος εκπαιδευσης ηταν απαγορευτικος στα πειραματα με τις διαφορετικες παραμετρους επιλεχτηκε ένα υποσυνολο με 400 εικονες.

- **<u>Πείραμα 2ο - Number of encoder/decoder layers:</u>**

Ας ξεκινησουμε βλεποντας καποια πειραματα που δεν ηταν καλα και τα συμπερασματα που προκυπτουν από αυτά. Δοκιμαστηκε να αλλαχθει στην αρχιτεκτονικη ο αριθμος τον encoders και ο αριθμος των decoders. Παρακατω βλεπουμε το loss και το mAP αυξανοντας από 6 σε 8 των αριθμο για τους encoders και τους decoders και βλεπουμε ότι το Map Μετα από 100 Epochs είναι πολύ κακο(10%). Στο επισημο paper υπηρχαν που πειραματα που με την αυξηση των encoders/decoders βελτιωναν τα αποτελεσματα εδώ αυτό δεν συμβαινει γιατι εχουμε ένα υποσυνολο από το dataset ενώ στο paper χρησιμοποιουσαν την Coco.

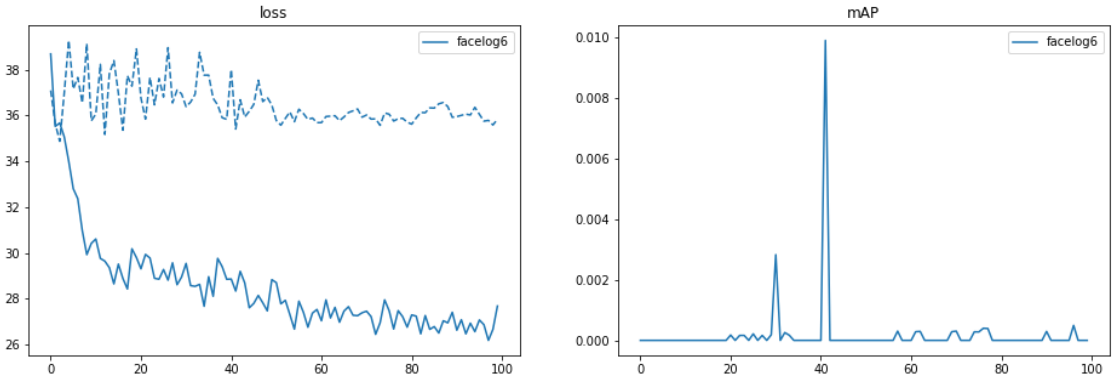


Παρακατω βλεπουμε για 3 encoders layers και 1 decoder και βλεπουμε ότι ενώ το loss είναι χαμηλο σε σχεση με πριν το mAP παλι δεν είναι καθολου καλο σχεδον 0.



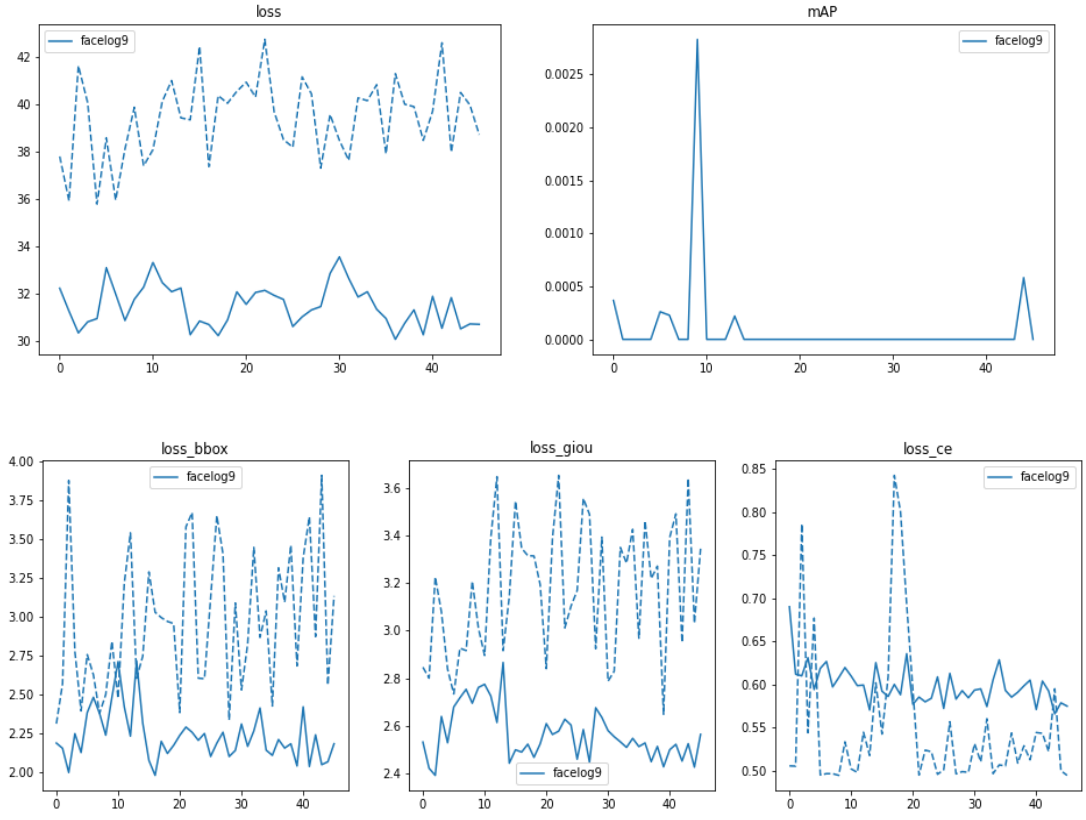
- **Πείραμα 3ο - Training from scratch:**

Το να εκπαιδευσουμε from scratch με 400 εικονες δεν ηταν καλη ιδεα και είναι και λογικο όπως φαινεται παρακατω. Αν χρησιμοποιουσαμε ολοκληρο το dataset θα μπορουσαμε να το κανουμε αφου θα ειχαμε τουλαχιστον 10000 εικονες, για αυτό χρησιμοποησαμε pre-trained βαρη από την Coco και καναμε fine-tuning.



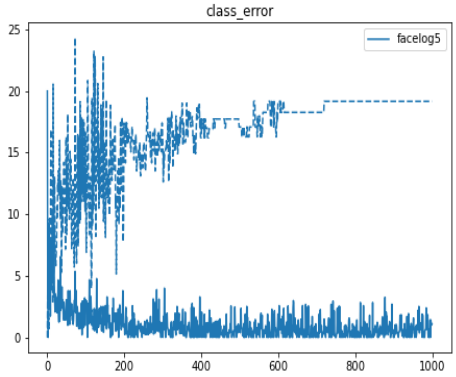
- **Πείραμα 4ο - μεγάλο learning rate=10-3:**

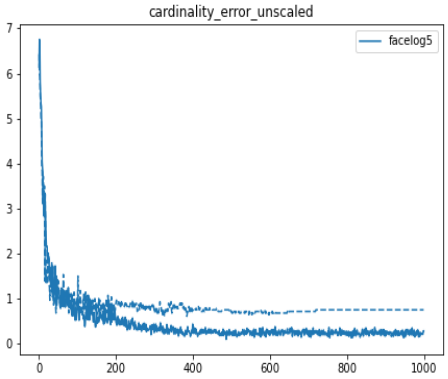
Εδώ βλεπουμε άλλο ένα κακο πειραμα που χρησιμοποιηθηκε μεγαλο learning rate και από ότι φαινεται αυτό δεν βοηθησε καθολου το δικτυο,πιεζοντας δηλάδη να μαθει γρηγορα δεν εχει τα επιθυμητα αποτελεσματα.

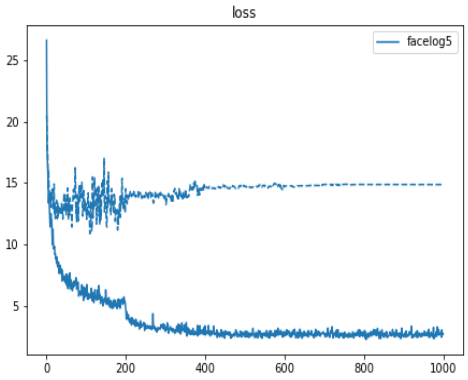


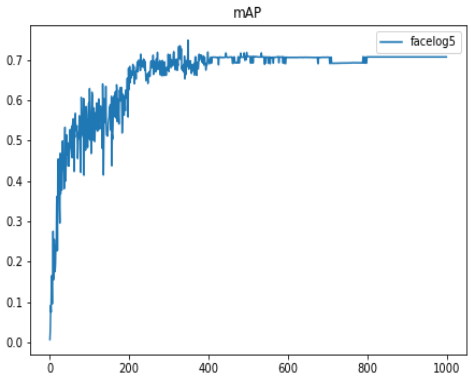
- **<u>Πείραμα 5ο – Training 1000epochs:</u>**

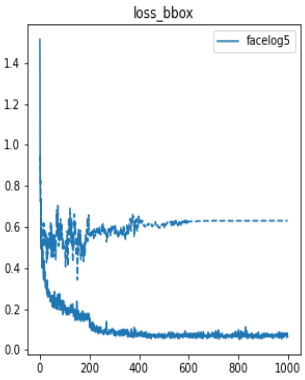
  Βλεπουμε εδώ ένα πειραμα για 1000 epochs που το Overfitting γινεται πολύ Εντονο μετα τα 150 epochs.

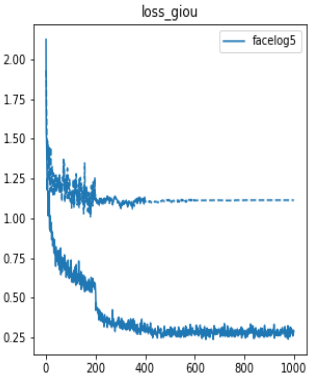


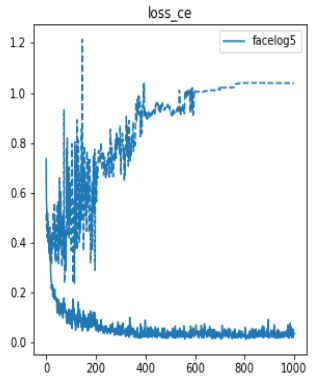






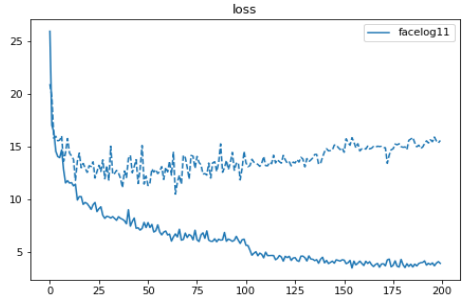


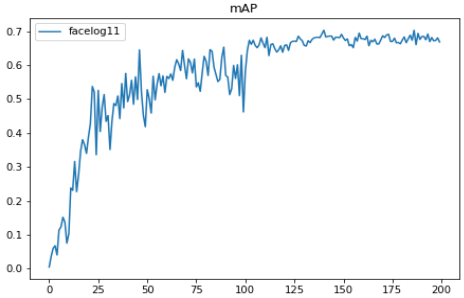


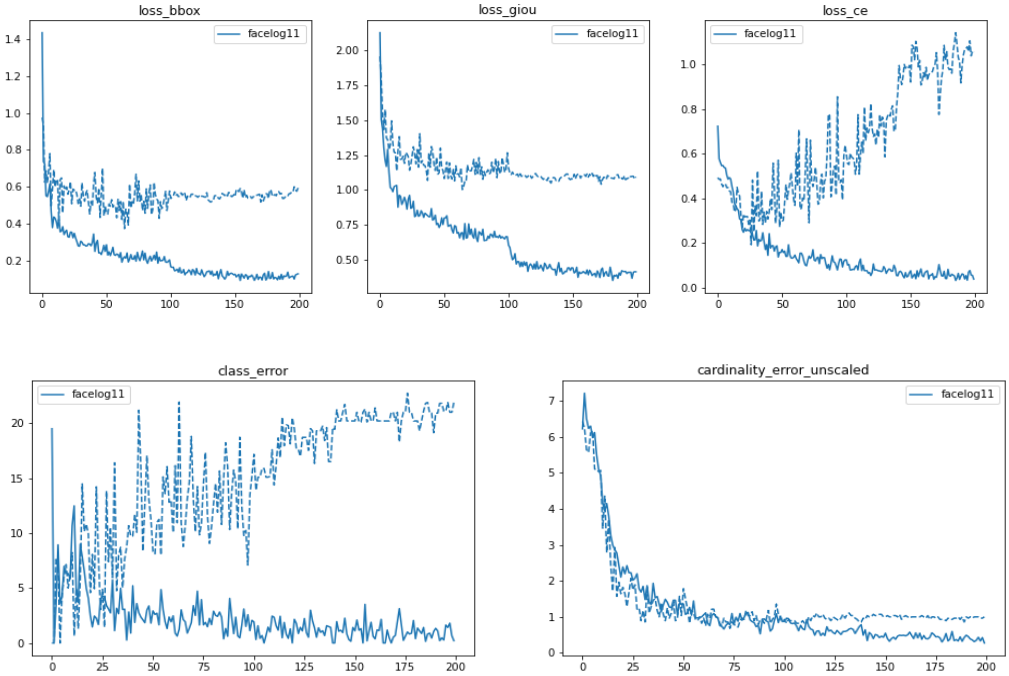


- **<u>Πείραμα 6ο – Training 200epochs:</u>**

Αναλογη συμπεριφορα βλεπουμε και για 200 epochs σε σχεση με πριν



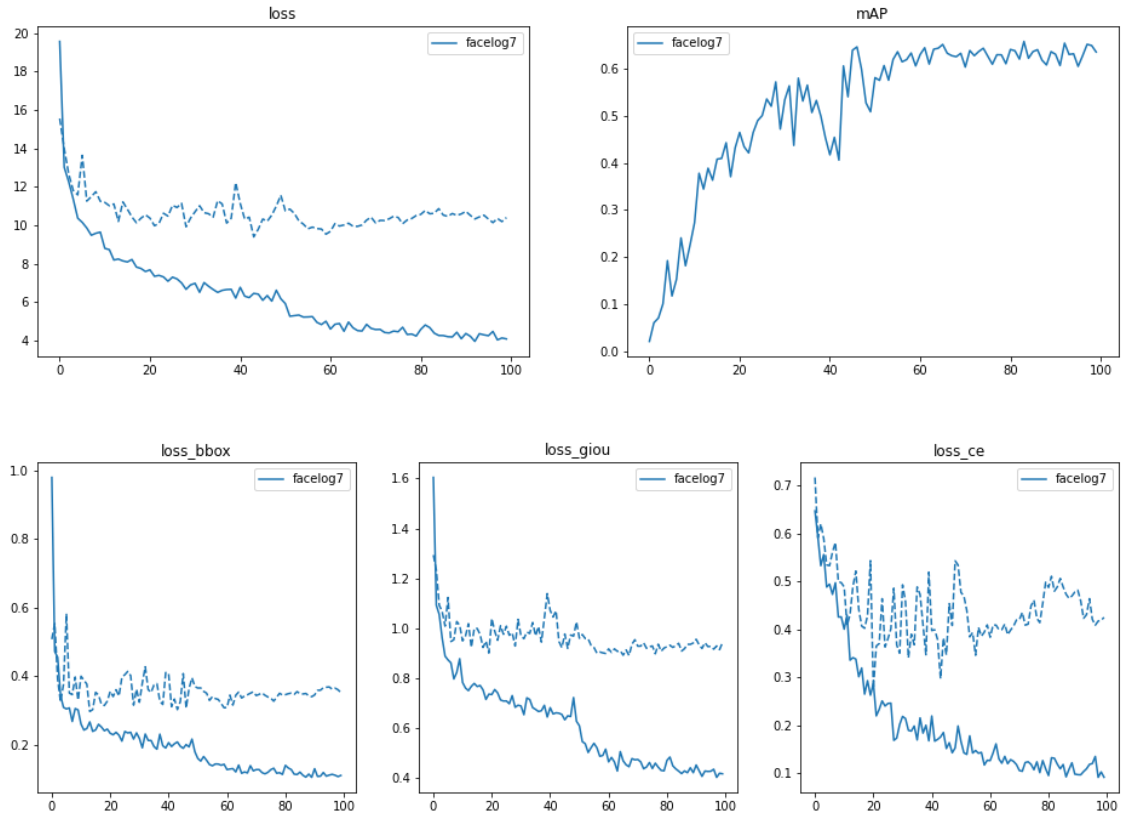




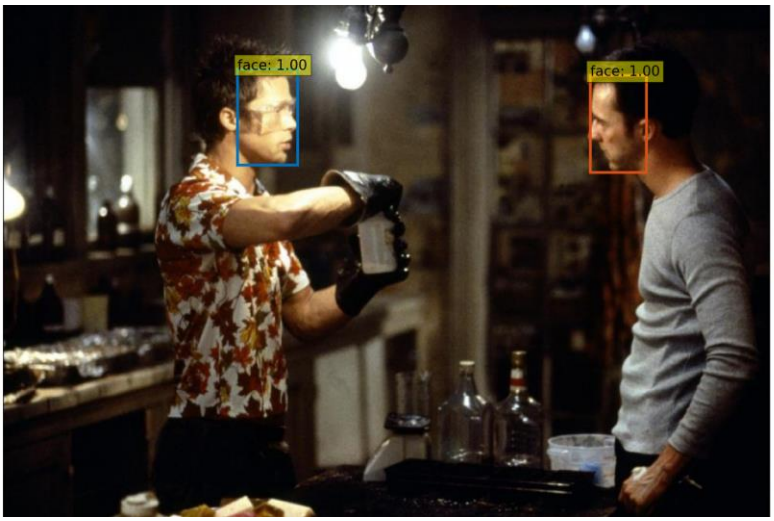
Θα μπορουσαμε να το αντιμετωπισουμε εκπαιδευοντας με παραπανω δεδομενα.

- **<u>Πείραμα 7ο – Training Number of queries=50:</u>**

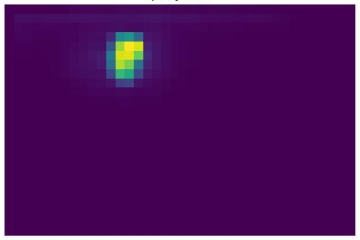
  Παρατηρουμε εδώ ότι αυξανοντας τα queries από 10 σε 50 το loss πηγαινει καλυτερα.



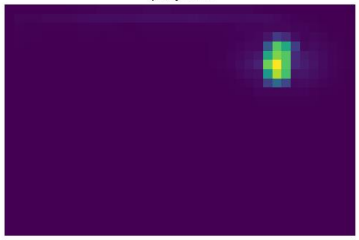
5. Visualizing predictions - Visualize encoder-decoder multi-head attention weights

- 

query id: 1



query id: 5



- 

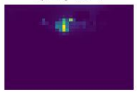
query id: 0

query id: 1

query id: 2

query id: 3

query id: 4



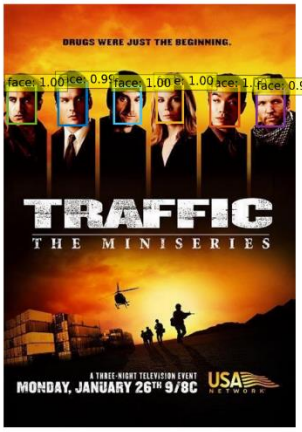
query id: 5

query id: 6

query id: 7

query id: 8

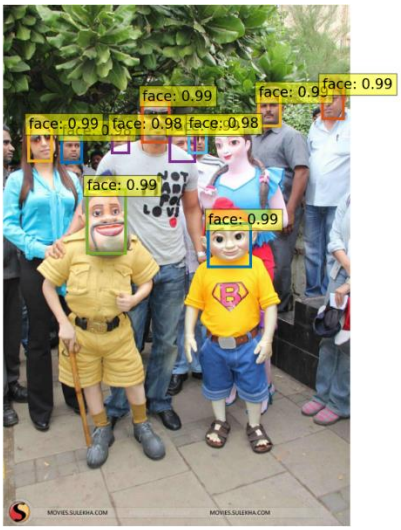
query id: 9

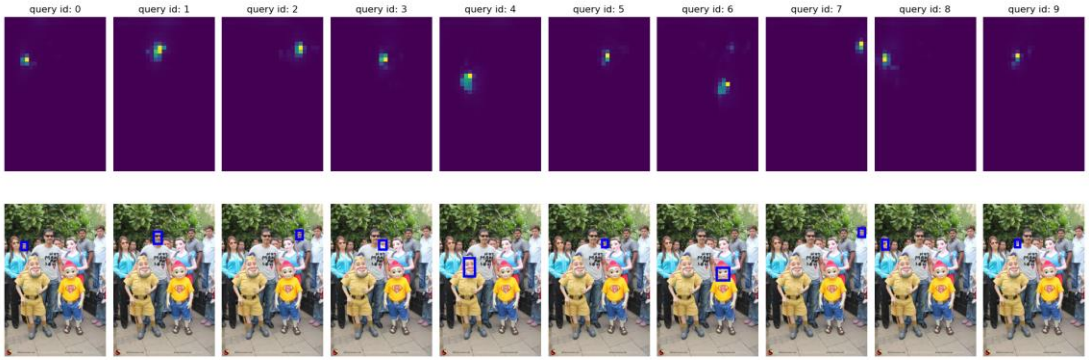

DRUGS WERE JUST THE BEGINNING.
face: 1.00
face: 0.99
face: 1.00
face: 1.00
face: 0.99
TRAFFIC
THE MINISERIES
A THREE-NIGHT TELEVISION EVENT
MONDAY, JANUARY 26TH 9/8C
USA NETWORK


query id: 0
query id: 2
query id: 5
query id: 7
query id: 8
query id: 9

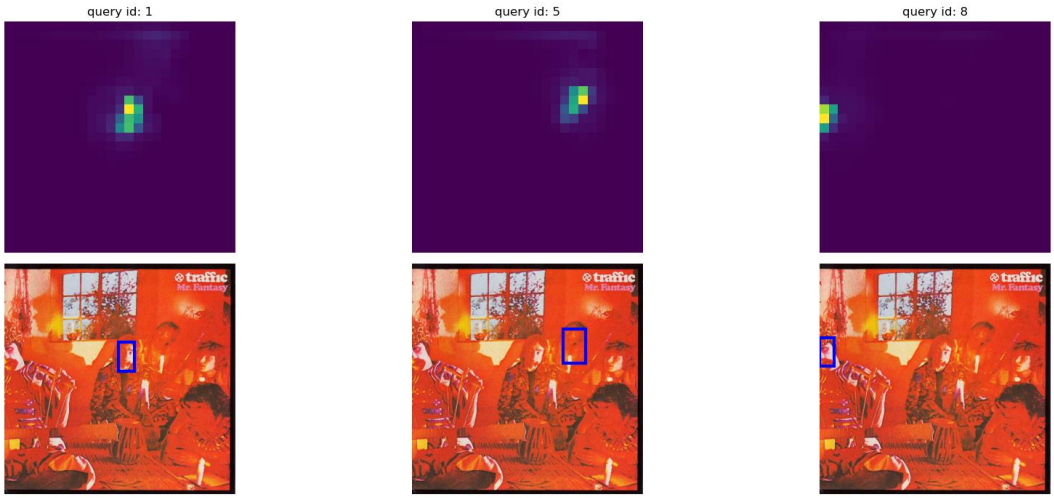

face: 0.99
face: 0.99
BALL
FOX 5 ATLANTA
FOX 5
myfoxatlanta.com
5:18 82°
NEW GEORGIA MILLIONAIRES

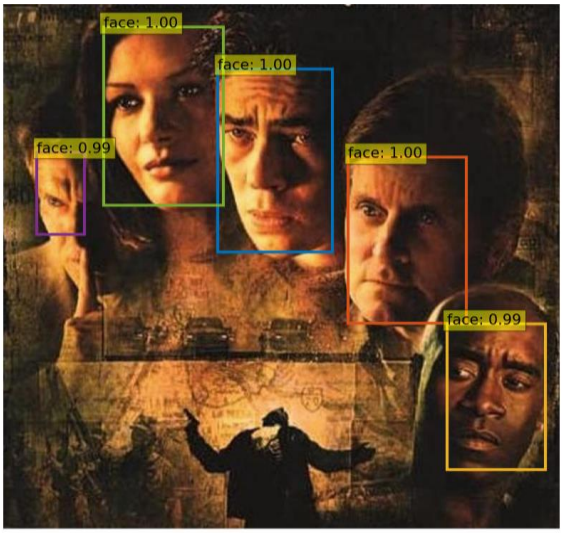

query id: 1
query id: 9


face: 0.99
face: 0.99
face: 0.99
face: 0.99
face: 0.98
face: 0.98
face: 0.99
face: 0.99
MOVIES.SULEKHA.COM
MOVIES.SULEKHA.COM


query id: 0
query id: 1
query id: 2
query id: 3
query id: 4
query id: 5
query id: 6
query id: 7
query id: 8
query id: 9


traffic
Mr. Fantasy
face: 0.51
face: 0.92
face: 0.52


query id: 1
query id: 5
query id: 8
traffic
Mr. Fantasy
traffic
Mr. Fantasy
traffic
Mr. Fantasy

- 
face: 1.00
face: 1.00
face: 0.99
face: 1.00
face: 0.99


query id: 1
query id: 5
query id: 6
query id: 8
query id: 9

- 
face: 0.93
face: 0.97


query id: 0
query id: 5

6. Επίλογος

Στην εργασία αυτήν χρησιμοποιήθηκε μια state-of-the-art Αρχιτεκτονική η οποία προτάθηκε μέσα στο 2020 και εισάγει transformers σε ένα κλασσικό πρόβλημα του Computer Vision όπως είναι το Object Detection, σημαντικό πλεονέκτημα της είναι η απλότητα της και ότι δεν υπάρχουν πλέον τα Anchors & Non-maximum suppression components που υπάρχουν σε κλασσικές αρχιτεκτονικές για object detection(Mask R-CNN, Faster κλπ.) και πρέπει να γίνει σωστό tuning των παραμέτρων τους. Λύσαμε ένα Face Detection πρόβλημα, χρησιμοποιήθηκε ένα custom Dataset με πρόσωπα και έγινε fine-tuning ξεκινώντας με pretrained βάρη από την COCO. Το dataset έπρεπε να έρθει σε coco format, να αλλάξει ο data loader και να γίνουν και αλλαγές σε αλλά σημεία στον αρχικό κώδικα ώστε να μπορεί να χρησιμοποιηθεί για το πρόβλημα του face detection. Σημαντικός χρόνος αφιερώθηκε στην μελέτη του αρχικού paper (DETR: End-to-End Object Detection with Transformers) αλλά και στην κατανόηση των Transformers. Επιπλέον έγιναν αρκετά πειράματα αλλάζοντας παραμέτρους όπως είναι το learning rate, learning rate drop, τον αριθμό των encoders και decoders της αρχιτεκτονικής, τον αριθμό των number queries κλπ. και προσπαθώντας κάθε φορά να ερμηνευτεί το αποτέλεσμα των αλλαγών μας. Παρουσιάστηκαν ακόμα οπτικά αποτελέσματα με της προβλέψεις του μοντέλου αλλά και τα attention weights του τελευταίου layer του decoder και είδαμε ποια pixel στην εικόνα ήταν αυτά που οδήγησαν το μοντέλο να κάνει μια συγκεκριμένη πρόβλεψη.