**Assignment4 - Facial Manipulation Detection**

**Bari Arviv - 308537034**

**Tomer Shimshi - 203200480**

**Question 1:** Done.

**Question 2:** Graphical user interface, application

Description automatically generated

**Question 3:** Done.

**Question 4:** Done.

**Question 5:** Done.

**Question 6:**

אכן המידע המתקבל מן הjson הגיוני, ניתן ליראות כיצד מדד הaccuracy עולה מepoch לepoch מ49% בהתחלה ל 78% בסוף (עבור הvalidation accuracy) ומאידך הloss יורד מ0.00613 ל0.00359 בסוף (עבור הvalidation loss) כלומר כמעט נחתך בחצי. (כמובן שגם בשאר ערכי הaccuracy והloss ישנה ירידה ערכי הvalidation נבחרו כדוגמה)

**Question 7:**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Question 8:**

The highest validation accuracy we got was 0.804 (corresponding to the 4th epoch) and the corresponding test accuracy is 0.757

**Question 9:**

In the test dataset there are 700 fake images and 1400 real images. But we constructed the length of the DB to be the min length of the real/fake DB so it reflects a DB that is equal in size.

**Question 10:**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Question 11:**

The graphs show completely different results because:

**Question 12:**

Done.

**Question 13:**

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

**Question 14:**

The highest validation accuracy is 0.536 and the corresponding test accuracy is 0.515

**Question 15:**

The proportion between the real images and fake images is 551(real)/552(fake)

But the way we implemented the DB is that we have equal number of samples per DB

**Question 16:**

We got a 50 50 classifier which basically means that our labeling is equal to a coin flip.

Or in other words the synthetic dataset managed to completely fool our SimplNet.

Chart, line chart

Description automatically generated ניתן לראות מתוך התמונה בסעיף זה, שהגרפים מראים שהמודל כושל. לדוגמא, ניתן להבחין בערכו של ה- הוא 0.5. כאשר ערך ה- הוא 0.5, ניתן לומר שמודל יבצע ניחוש אקראי, מתחת לערך זה, המודל נחשב כושל ומעל לערך זה, ניתן לומר שהמודל מסוגל ללמוד תכונות על ה-. במקרה זה, הערך בדיוק 0.5, כלומר, המודל בוחר בצורה אקראית סיווג. לעומת זאת, במקרה בו המודל אומן על , ערך ה- הוא 0.842, כלומר, המודל טוב וביצועיו טובים בהתאם. בנוסף, לאחר שבחנו את התמונות ב- ראינו שהן לא נראות איכותיות בהשוואה

ל-, כמו כן, כמות ה- שניתנה בין שניהם, שונה לחלוטין בכמותה וזה משפיע באופן ישיר על ביצועי המודל גם כן.

**Question 17:**

We can clearly see that the Fake Dataset doesn’t look like real people images while the Synthetic Dataset looks a lot like real people images

**Question 18:**

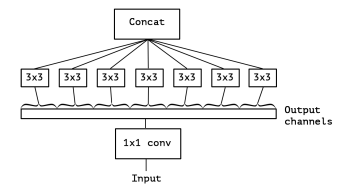
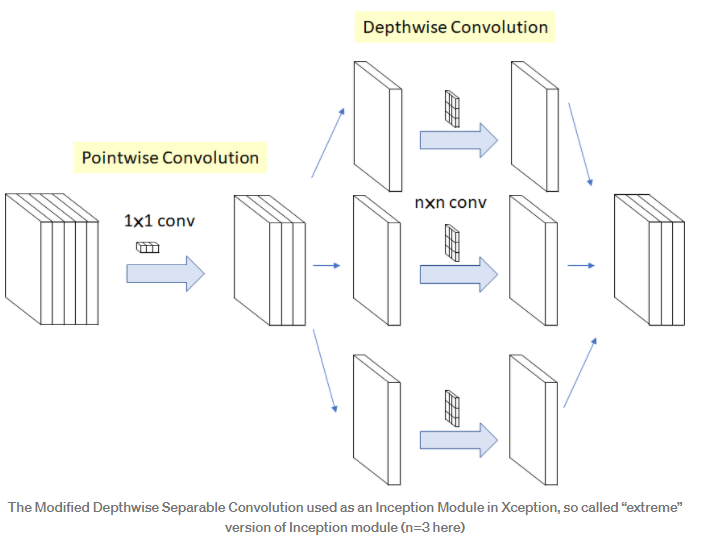
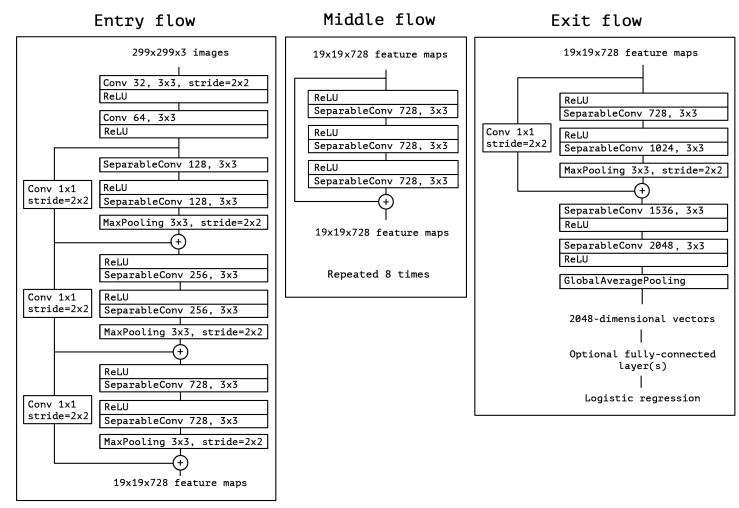
Xception is pre-trained on the ImageNet Dataset, which is consist of 14197122 annotated images we can get all the papers related to ImageNet on [this link](https://paperswithcode.com/dataset/imagenet)

**Question 19:**

ארכיטקטורת היא ערימה ליניארית של שכבות קונבולציה הניתנות להפרדה בעומק עם חיבורים שיוריים. זה הופך את הארכיטקטורה לקלה מאוד להגדרה ולשינוי.

אבני הבניין הבסיסיות של מודל הן:

1. : השכבה יוצרת מפת תכונות על מנת לחזות את ההסתברויות של המחלקה עבור כל תכונה ע"י גרעין/מסנן הסורק את כל התמונה (על פי גודלו והצעד המוגדרים).
   * : בעיה ידוע ברשתות עצביות קונבולוציוניות עמוקות היא שמספר מפות התכונות גדל לעיתים קרובות עם עומק הרשת, בעיה זו עלולה לגרום לעלייה דרמטית במספר הפרמטרים. כדי לטפל בבעיה זו משתמשים בשכבה זו המציעה , לאחר שכבה זו, נצמצם באופן משמעותי את הממד מבחינת עומק. השימוש בה מאפשר להפחית ממדיות, להקטין את מספר מפות התכונות תוך שמירה על התכונות הבולטות שלהן.
2. : פונקציה לא ליניארית, , אחת מהנפוצות ביותר בתוך השכבות הנסתרות של רשתות עצביות עמוקות.
3. : שכבה זו מבצעת הפרדה בעומק, . ראשית, פועלת על כל ערוץ קלט בנפרד, שנית, מבצעת ערבוב של כל התוצאות ולבסוף, מוציאה כפלט תוצאה הכוללת את כל הערוצים. שכבה זו יעילה מבחינת עלות החישוב וזיכרון. לדוגמא: עבור מסנן בגודל , קלט בעל 3 ערוצים ו-64 מסננים, המספר הכולל של המסננים - עבור שכבה זו - , לעומת זאת, עבור שכבת קונבולוציה רגילה (סעיף 1) - .
4. : שכבה זו משמשת להפחתת הכמות הכוללת של הפרמטרים ברשת וכן להפחתת המורכבות, ובכך מאיצה את החישוב ותורמת להימנע מהתאמת יתר . שכבה זו שומרת על הערך הגדול ביותר.
5. : שכבה זו נועדה להחליף שכבות מחוברות לחלוטין במודלי . הרעיון הוא ליצור אחת עבור כל מחלקה מתאימה של משימת הסיווג מהשכבה הקודמת לה ולאן מכאן, מבצעת מיצוע.



**Question 20:**

Same as Q18

**Question 21:**

The input feature dimensions for the final classification block ”fc” is 2048 dimensional vectors taken from Xception original [paper](https://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf)

**Question 22:**

The number of parameters the network holds by default is 22855952 according to the Xception original [paper](https://openaccess.thecvf.com/content_cvpr_2017/papers/Chollet_Xception_Deep_Learning_CVPR_2017_paper.pdf) and from the function utils.get\_nof\_params we get the same number of parameters 22855952.

**Question 23:**

Done. And changed the model to save the pretrained model weights and only trained the FC layer.

**Question 24:**

We have added to the Xception parameters count

**Question 25:**

Done.

**Question 26:**

The received graphs were:

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

As you can see we go a bit of an overfitting, usually this problem could be solved this problem by adding dropout layers to the model, but we still got much better results than the SimpleNet

**Question 27:**

The highest validation accuracy is 0.917 and the corresponding test accuracy is 0.907

And due to overfitting they correspond with first epoch

**Question 28:**

Done

Chart

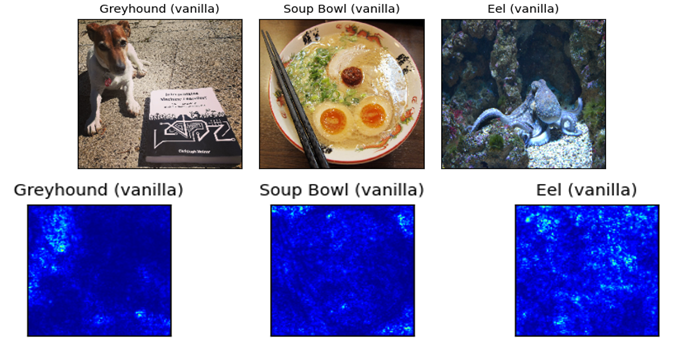
Description automatically generated

Chart, line chart

Description automatically generated

**Question 29:**

Image-Specific Class Saliency מבצע חישוב של הגרדיאנט של פונקציית העלות עבור המחלקה בה אנו מעוניינים ביחס לפיקסלים של הקלט ומתקבלת מפה כגודל תכונות הקלט עם ערכים. כלומר, הגישה מבצעת, forward pass עבור תמונת הקלט ומחשבת את הגרדיאנט של score המחלקה ביחס לפיקסלים שהתקבלו כקלט. לעיתים קרובות, מודלי CNN משתמשים בפונקציית ReLU ולכן, כל הערכים השליליים מתאפסים, ReLU=Max(0,X\_n ). כתוצאה מכך, כאש פונקציית ההפעלה של הנוירון היא אפס, איננו יכולים לדעת מהו ערכו המקורי. במקרה של Vanila Gradient, הבעיה נפתרת באופן הבא: שיטה זו לוקחת את הגרדיאנט שהופץ עד כה מהשכבה העוקבת ומגדיר את הגרדיאנט לאפס כאשר ערך פונקציית ההפעלה הוא שלילי. לשיטה זו יש בעיית רוויה משום שהיא מוגדלת בערכיה מ-0 ומעלה ולכן, אם כל הערכים שליליים, הם כולם יתאפסו ולא יהיו שינוי כלל.



**Question 30:**

היא טכניקה להפקת הסברים חזותיים להחלטות של מודלים מבוססי . השיטה מבצעת מיפוי מחלקות משוקלל, באמצעות הגרדיאנטים של כל מחלקות היעד , דרך שכבת הקונבולוציה האחרונה במטרה לייצר מפת לוקליזציה המדגישה אזורים חשובים בתמונה. השיטה מקצה לכל נוירון ציון רלוונטיות להחלטת חיזוי המחלקה, במטרה להבין באילו חלקים בתמונה שכבת הקונבולוציה "מחפשת" סיווג מסוים. צריך להחליט עד כמה חשובה כל אחת ממפת התכונות של ה- למחלקה ע"י שקלול כל פיקסל עבור כל תכונה ב- באמצעות הגרדיאנט ולאחר מכן, מתבצע ממוצע על פני כל ה-. דבר זה, מספק מפת חום שמדגישה אזורים המשפיעים באופן חיובי או שלילי על החיזוי. ערכי המפה מועברים דרך פונקציית , כלומר, כל הערכים השליליים מתאפסים, , הסיבה לכך היא שאנחנו מעוניינים רק בחלקים שתורמים למחלקה שנבחרה ולא למחלקות אחרות. ישים למגוון רחב של משפחות מתוך מודלי ללא כל צורך בשינוי הרכב המודל או אימונו מחדש. במקרה בו מדובר מודל לסיווג תמונות, השיטה מאפשרת ליצור ויזואליזציה נפרדת עבור כל מחלקה קיימת בתמונה, מעניקה תובנות לגבי ביצועי המודל, תורמת להשגת הכללה על ידי זיהוי מערך נתונים בעל הטיה ועוד.

תמונה שמכילה טקסט, כלב, שונה, חתול

התיאור נוצר באופן אוטומטי

**Question 31:**

Done.

**Question 32:**

The figure we get after running the command: python saliency map.py -m XceptionBased.cpp checkpoints/synthetic dataset XceptionBased Adam.pt -d synthetic dataset is: A picture containing graphical user interface

Description automatically generatedChart

Description automatically generated

The figure we get after running the command python saliency map.py -m SimpleNet -cpp checkpoints/fakes dataset SimpleNet Adam.pt -d fakes dataset is:

A picture containing graphical user interface

Description automatically generated

Chart

Description automatically generated

**Question 33:**

Done.

**Question 34:**

Done.

**Question 35:**

The figure we got after running

python grad\_cam\_analysis.py -m SimpleNet -cpp checkpoints/fakes\_dataset\_SimpleNet\_Adam.pt -d fakes\_dataset is:

Surface chart

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

a

The figure we got after running

python grad\_cam\_analysis.py -m SimpleNet -cpp checkpoints/synthetic\_dataset\_SimpleNet\_Adam.pt -d synthetic\_dataset is:

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

The figure we got after running

python grad\_cam\_analysis.py -m XceptionBased

-cpp checkpoints/synthetic\_dataset\_XceptionBased\_Adam.pt

-d synthetic\_dataset is:

Graphical user interface

Description automatically generatedGraphical user interface

Description automatically generated

**Bonus part:**

עבור חלק הבונוס בחרנו להשתמש בתור בסיס ברשת מאומנת מראש [resnet18](https://paperswithcode.com/lib/torchvision/resnet) על רשת זו הוספנו בכניסה שכבת Upsample בכדי להביא את הכיסה לגודל הרצוי בכניסה resnet בהתחלה בכדי להקל על למידת המערכת הקפאנו את המשקולת של המודל המאומן (אשר במוצאו אנו מקבלים שכבת fully connected באורך 1000 ) אך נוכחנו כי במקרה זה הרשת לא מקבלת תוצאות מספיק טובות החלטנו שכן לאמן שכבות אלו) ועל שכבות המודל הרגילות הוספנו שכבות של נרמול, שכבות fully connected ושכבות של dropout בכדי למנוע מצב של overfitting לאחר מכן אימנו מודל זה על הtrain dataset של fakes\_dataset ולבסוף קיבלנו 91% accuracy על dataset זה.