

Image Processing - Exercise 5

Agam Hershko, id_214193831, 214193831

מבוא

מטרת התרגיל היא לשחזר תמונה באמצעות **Inversion GAN** תוך שימוש ברשת **StyleGAN2**. הטכניקות המרכזיות שהשתמשו בהם בתרגיל הן אופטימיזציה של ה-**latent vector** – במקום לעדכן את משקל הרשת, אנו מעדכנים את ה-**latent vector** כדי למזער את ההבדל בין התמונה שנוצרה dazu המוקורית. כמו כן, מתבצע שימוש ברשומות גנרטיביות כ-**Image Priors**: הרשת **StyleGAN2** אומנה על תמונות פנים, ומונצלת כדי לשחזר תמונות באופן קרוב יותר למציאות, תוך ניצול הידע שהרשת צברה במהלך האימון. הכוונה היא שהתשומות שהרשת מסוגלת ליצור נחבות כ"בתוכה תחום הידע שלה", בעוד שהתשומות שהיא לא מצליחה לשחזר נחבות כ"מחוץ לתחום".

החלק המרכזי שהיינו צריכים למשב בתרגיל הוא מימוש הפקתה לתמונה (**degradation**) מסוגים שונים – למשל טשטוש גאוסי (**blur**), הסרת צבע (מעבר מתמונה צבעונית ל-**grayscale**) והפעלת מסכה קר שיהיא חלק חסר בתמונה (מופייע מלבן שחור). המטרה היא לשחזר את התמונה כך שההפקתה לא תהיה ניכרת. נעשה זאת זה על ידי הפעלת פונקציית הפקתה על התמונה שיוצרה (**the generated image**) לפני חישוב פונקציית הפסד (**loss**). באמצעות הטכניקות שתוארו ניתן לשחזר תמונות באיכות גבוהה ולשפר תמונות שעברו הפקתה על ידי שימוש בידע שהרשת צברה במהלך האימון.

אלגוריתם

האלגוריתם מבצע אופטימיזציה על ה-**latent vector** של רשת **StyleGAN2** כדי לשחזר תמונה נתונה, תוך שימוש בפונקציית הפסד (**loss**) המודדת את ההבדלים בין התמונה המתקבלת לתמונה המקורית. במהלך האופטימיזציה, הרשת משתמשת גם בקביעת רגולרייזציה על ה-**latent vector** כדי למנוע רעש ושינויים קיצוניים בתוצאות. **GAN** הוא מודל גנרטיבי הכליל רשת גנרטור שמתפרקת לייצור תמונות חדשות, ורשת דיסקרימינטור המבחןינה בין תמונות אמיתיות ומיוזיפות. **StyleGAN2**-ב-**latent vector** הוא וקטור קבוע שמייצג את המאפיינים הגלויים של התמונה, ומשפיע על יצרתה על ידי הגנרטור, תוך שימוש בחילוק ובתוקן. במהלך האימון, הרשת מתאימה את ה-**latent vector** כך שהתמונה הגנרטית שתיווצר תהיה קרובה ככל האפשר לתמונה המקורית, תוך שמירה על סדר פעולות חשוב לשבור מדיוק עם מינימום שגיאות.

האלגוריתם של **StyleGAN2 invert_image** מבצע אופטימיזציה על ה-**latent vector** של רשת **StyleGAN2** במטרה לשחזר תמונה נתונה. בהתאם, תחילה טובעים את הרשת ואת תמונה המטרה, כאשר התמונה עוברת חיתוך והתאמה לרוחוציה של הרשת. לאחר מכן, האלגוריתם מבצע אופטימיזציה על ה-**latent vector** כדי למזער את ההבדל בין התמונה המתקבלת לתמונה המקורית, תוך שימוש ברגולרייזציה על ה-**latent vector** כדי למנוע רעש ושינויים קיצוניים בתוצאה. במהלך האופטימיזציה, נשמרות תמונותBINים שמציגות את השיפור בתהילך, ולבסוף נשמר ה-**latent vector** הסופי והתמונה המתקבלת.

הfonקציה latent_optimization על StyleGAN2 מבצעת אופטימיזציה של ה-z latent vector על מנת לשחרר תמונה מטרה. היא מתחילה בחישוב סטטיסטיות של ה-space latent, כולל ממוצע וסטיית תקן, ובמבצעת אופטימיזציה באמצעות שימוש במודל VGG16 להעריך את השגיאה בין תכונות התמונה המתקבלת לתמונה המטרה. האופטימיזציה כוללת שלושה מרכיבים עיקריים בהפסד: הפסד פרספקטיבי (Perceptual Loss) שמודד את ההבדלים בתכונות, רגולציה על רעש שמחזיקה את השפעת הרעש בתמונה המתקבלת, ורגולציה על latent vector כדי לשמור אותו קרוב לממוצע. הfonקציה משתמשת ב-optimizer Adam ומבצעת שדרוגים לאורך מספר שלבים עד להשגת אופטימיזציה מרבית.

הfonקציית LOSS מוחשבת את סך השגיאה בין התמונה המתקבלת והמטרה, עם שילוב של Loss Perceptual ו-Loss Regularization, שמחזיכים רעש ומבצעים אופטימיזציה יציבה על latent vector. פונקציית ההפסד הותאמת להפחיתה בתמונה כמו שתואר מקודם.

3.1 תוצאות

בחלק הנוכחי יוצגו התוצאות לפני ואחרי יישור תמונה. תהליך היישור בוצע על התמונה באמצעות תהיליך האימון של StyleGAN2 על FFHQ. היישור מבטיח שהחלקים של הפנים ממוקמים במיקומים אחידים בתמונה, כך שההתמונה תהיה עקבית עם האופן שבו אומנה הרשות. בתמונות שלפני ואחרי ניתן לראות את השיפור במיקומן של הפנים לאחר היישור. התמונה השמאלית היא טרם הליך היישור ובها ניתן לראות גם את החלק העליון של החזה, הצוואר וחלק גדול יותר מהרפק ולוונת ניתן להבחן בתמונה הימנית שבתמונה מופיעים הפנים בעיקר ללא חלקים לא רלוונטיים וחלק הפנים ממורכדים וממוקמים בתמונה.



3.2 תוצאות

בחלק הנוכחי נציג את התוצאות לפני ואחרי הפעלת Inversion GAN. בחלק זה לא מתבצע הפחתה של התמונה (loss degradation).

השתמשתי בתמונה בכל התרגיל (כך הבנתי מההוואות). במסגרת תהליך GAN Inversion, המטרה הייתה לשחרר תמונה נתונה על ידי אופטימיזציה של ה-z latent vector כך שההתמונה הganרטיבית תתאים כמה שיותר לתמונה הקלט.

לאחר מכן, הרצתי את הקוד עם תמונה שנבחרה והתחלתי לעקוב אחרי תהליך האופטימיזציה. שיחקתי עם פרמטרים כמו מספר הצעדים (latent_steps) ומשקל הרגולציה של המרחק מה-z latent vector.

המוצע (`latent_dist_reg_weight`). פרמטרים אלה משפיעים על המרחק שבו ה-*vector* יכול להיות מהממוצע של התמונות שהגנרטור אומן עליהם ומשפיעים ישירות על איכות השחזור. התוצאה הייתה ש恢 Zur מדיוק של התמונה, כאשר ראייתי את השפעת השינויים בפרמטרים על תהליכי האופטימיזציה. להלן מציג את התוצאות הייזואליות ואסבייר אודוטיהן ולאחר מכן אדון בפרמטרים.

התמונה המקורית



התמונה לאחר 200 צעדים



התמונה לאחר 0 צעדים



התמונה לאחר 300 צעדים



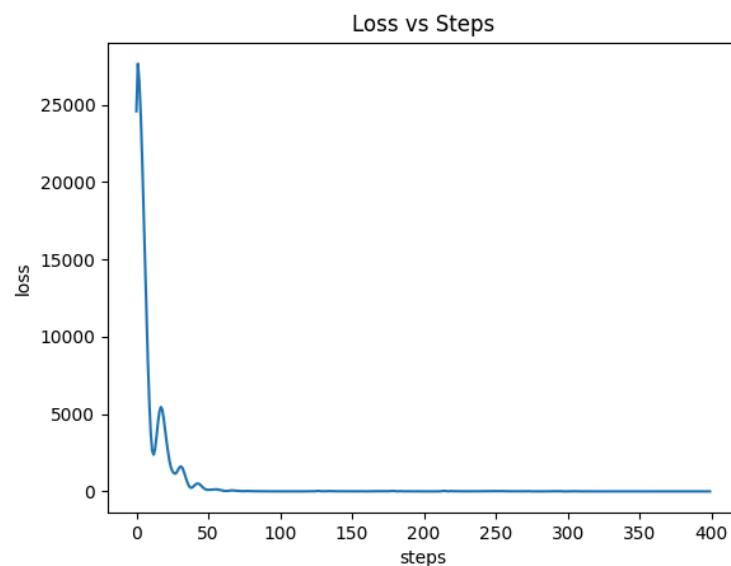
התמונה לאחר 100 צעדים



התמונה הסופית (לאחר 400 צעדים)



ניתן לשים לב שלאחר 200 צעדים התמונה מתיצבת על התוצאה הסופית. להלן יוצג גרף של פונקציית ההפסד כתלות במספר הצעדים. ניתן להבחין בהתייצבות הפונקציה בגרף.



מספר הצעדים (`steps`) מופיע על איקות התוצאה הסופית בצורה משמעותית. הפרמטר `steps_num` קובע את מספר הצעדים בתהליך האופטימיזציה, כלומר את כמות העדכנים שעובר ה-*latent vector* במהלך חיפוש הווקטור המתאים ביותר לשחרור התמונה. מספר צעדים קטן מדי לא אפשר לקטור להתעדכן עד לתוצאה הרצiosa. מספר גדול מדי של צעדים יכול להוביל לעומס חישובי משמעותי, שכן כל צעד דורש עדכון נוסף של ה-*latent vector*, מה שמאיר את זמן האופטימיזציה ו מגביר את הצורך במשאבים חישוביים. מעבר לנחיצות של מספר צעדים אופטימלי, עשוי להיווצר בזבוז זמן ומושגים ללא שיפור ממשמעותי בתוצאה.

הפרמטר `latent_dist_reg_weight` קובע את המרחק של ה-*latent vector* מהמוצע. ערך קטן מדי לא אפשר לשחרר פרטימן מדויקים בתמונה, כי הווקטור לא יוכל לסתות מספיק כדי להتاימים לתמונה הקלט. מצד שני, ערך גדול מדי יגביל את הगמישות של הווקטור, יוכל להוביל לתוצאה שבה התמונה הגנרטיבית תהיה קרובה מדי למוצע של התמונות שהגנרטור אומן עליה (פרצוף גברי יותר).

3.3.1 תוצאות

להלן ציג את התוצאות של הטשטוש התמונה (blurring) כחלק מההפחטה (degradation) שעושים לתמונה המקורית ואת השחרור שעושים לתמונה תוך כדי ביטול הטשטוש (deblurring). כמו שהסבירתי בחלק של האלגוריתם, בפונקציה `invert_image` המבצעת `GAN Inversion` על תמונה נתונה נבקש קלט מהמשתמש כדי לדעת אם צריך להפעיל טשטוש גאותי אני טרם השחרור (יישום ההפחטה).

כדי לבצע את תהליך השחרור, נפעיל את הפונקציה `latent_optimization` שבין היתר מבצעת את ההפחטה - במקרה זהה טשטוש.

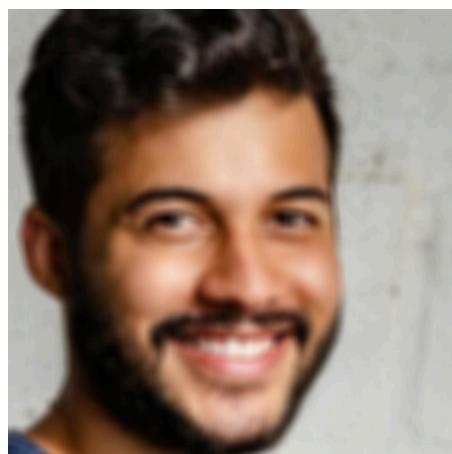
הlixir הטשטוש מתבצע על ידי שימוש בפונקציית `GaussianBlur` של ספריית `transforms`. גודל ה `kernel` הוא 25 וה `sigma` הוא 25. אלה הם `hyper parameters` שבעבורם לי כדי לקבל את אפקט הטשטוש הרצוי (כל שנגדיל את ה `kernel` נעשה מיותר לחלק גדול יותר של התמונה וככל שנגדיל את ה `sigma` רמת הטשטוש של התמונה גדלה). בהתחלה התמודדתי עם קושי כאשר לא

הדרתי את ה `sigma` ולכון לא היה ניתן לבדוק בטעות ממשמעותי בתמונה.

להלן התוצאות עבור התמונה שבחרתי:



Original

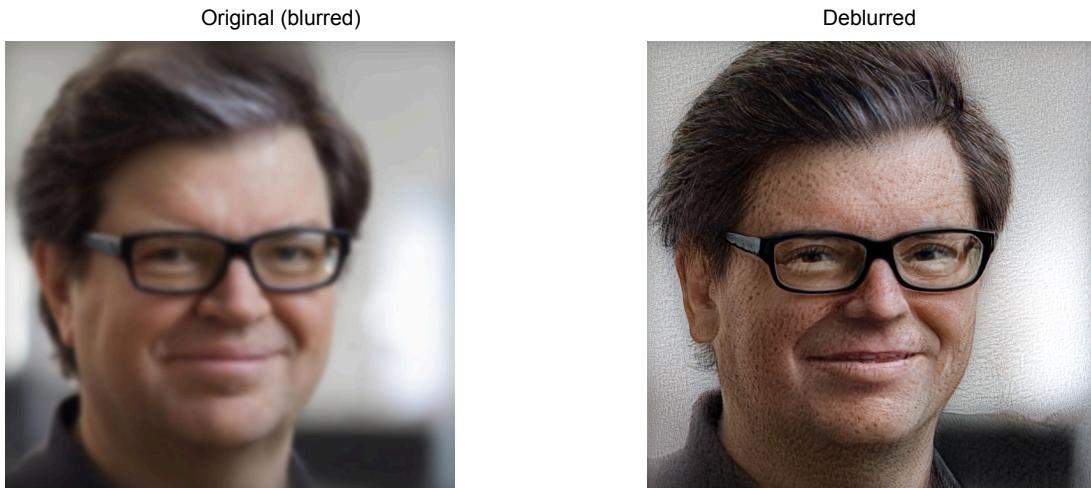


Blurred



Deblurred

להלן התוצאות עבור התמונה הננתונה (school example) של LeCun-Yann. כאשר התמונה שקיבliśmy היא כבר מטושטשת ולכון לא נטשטש אותה שוב בעצמנו לפני השחרור. השחרור בוצע בהצלחה.



תוצאות 3.3.2

להלן אציג את התוצאות של התמונה ב grayscale כחלק מההפקחה (degradation) שעושים לתמונה המקורית ואת השחזר שעושים לתמונה תוך כדי צביעה (colorization). ההפקחה משלבת כמו קודם transforms Grayscale וtorchvision עם פרמטר 3 כיוון הפורמט הדריש של ה grayscale הוא עם 3 ערוצים.
להלן התוצאות עבור התמונה שבחרתי (השחזר יצא עם פלטת צבעים קצת שונה אבל עדיין טוב):



להלן התוצאות עבור התמונה הנtypical (school example) של Alan Turing - כאשר התמונה שקיבliśmy היא כבר ב grayscale ולכן לא נפעיל את ההפקחה שוב לפני השחזר. התמונה המשוחזרת התבכעה בצורה טובה אך הרקע יצא יותר אדום במקור (התמונה הצבעונית לא סופקה אף מופיעה בהוראות).



3.3.3 תוצאות

להלן אציג את התוצאות של התמונה שהופעלה עליה מסכה כחלק מההפקחה (degradation) שעושים לתמונה המקורית ואת השחזר שעשויים לתקן תוך החלק החסר (inpainting). ההפקחה משולבת כמו מוקדם בקוד. ההפקחה עצמה ממומשת באופן הבא: נקראת המסכה לתוכן מערך ונמיר אותו ל tensor. כמו כן, נוצר מסכהBINARITY, נפעיל אותה על התמונה ונתחשב ב shifting הדורש.

להלן התוצאות עבור התמונה שבחרתי:



Original



Masked (degradation)

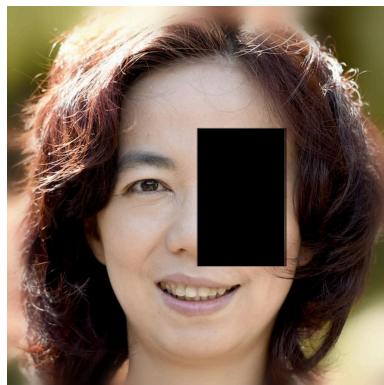


Impainted (reconstructed)

להלן התוצאות עבור התמונה הננתונה של Li Fei-Fei. נפעיל את ההפקחה והשחזר בוצע בהצלחה.



Original



Masked (degradation)



Impainted (reconstructed)

מסקנות

במהלך הפרויקט למדנו כי רשתות ניירונים יכולות לשמש ככלי חזק לשחזר ושיפור תמונות, תוך ניצול הידע שנוצר באימון מודלים גנרטיביים. נשים לב שזיהינו שהמודל תלוי בתנאים מקדים מסוימים, כמו מיקום וגודל הפנים. לכן, אחד המרכיבים הקritisטים בתהיליך היה יישור התמונות (alignment), אשר הבטיח שהפנים יוצגו בצורה תקנית כך שהרשת תוכל להתמודד איתן טוב יותר. בנוסף, תהליך ההיפר (inversion) אפשר לנו למצוא את latent vector הקרוב ביותר לכל תמונה, מה שאפשר לשחזר אותה בדיק מרבי. לבסוף, ראיינו כיצד שימוש בסוגי הפקחה שונים כמו טשטוש, הסרת צבעים או הסתרת חלקים מהתמונה משפיע על יכולת השחזר של המודל.