

# Image Processing - Exercise 5

Shir Rashkovits, shirashko, 209144013

## מבוא

מטרת התרגיל היא לבצע שימוש שחזור פנים לתמונה בעזרת מודל גנרטיבי תחרותי StyleGAN2. באופן ספציפי יותר, בהינתן גנרטור מאומן עם ידע מוקדים על תמונות פנים, ותמונה פנים כלשהי שעברה דגרדציה, למצוא את הוקטור הסמי שכאשר הוא ניתן כקלט לגנרטור, הפלט הוא התמונה הנתונה לפני הדגרדציה. כאמור, המטרה היא "לקודד" את התמונה לפני הדגרדציה לתוכה המרכיב הסמי של ה-GAN. לשם כך נשתמש במודול StyleGAN2, מודל גנרטיבי שאומן על תמונות פנים מגוונות, כך שיש לו prior knowledge על מאפייני תמונות פנים. ה prior הזה מחייב שלכל תמונה פנים טבעיות יש איזושה latent vector שכאשר הוא ניתן כקלט לגנרטור, קיבל את התמונה הרצiosa. בהינתן וקטור רעש רנדומלי, latent vector, הוקטור יוצר תמונה פנים כלשהי. לשם שימוש זאת, נבצע אופטימיזציה סמייה, בה הוקטור הסמי (כלומר הפלט של ה-GAN) עובר אופטימיזציה בעוד המשקלות של הגנרטור נשארות ללא שינוי. בשבייל האופטימיזציה הסמייה הניל ננצל את אלגוריתם ה- backpropagation המשמש ב- Gradient Descent כך שבכל שלב, הדיפרנציאלי מחושב ביחס לקלט לגנרטור, הקוד עצמו. כך נבצע Gan Inversion, חיפוש של הוקטור הסמי שהמරחק של פלט הגנרטור שעובר דגרדציה בהינתן, היא הקרובה ביותר לתמונה היעד שעברה דגרדציה. כאשר הקידוד של התמונה הוא בדיקוק הוקטור הסמי, נקבל את התוצאה המינימלית והאופטימלית, אפס. לשם כך נשתמש גם בעיקרון נוסף שראינו בכיתה שהוא perceptual loss, שמשתמש בפונקציה שסמיירה את התמונה לוקטור פיצרים במרחב הסמי, ולמעשה נגדיר את פונקציית loss על ידה.

## אלגוריתם

### 1. אתחול:

- טען את רשות StyleGAN2 שהוכשרה מראש, המכילה גנרטור שלמד לייצר תמונות פנים אנושיות.  
- קרא את תמונה היעד מהקובץ שצין ובצע preprocessing כדי שתתאים לצורה של המזופה מקלט לגנרטור.

- חחל את שיטת degradation המתאימה (פונקציית הזרחות, מעבר לגונו אפור, החלת מסיכה), על התמונה המעובדת כדי להכין אותה לשחזור.

- חשב נקודת התחליה לאופטימיזציה שתנסה את החיפוש אחר הוקטור הסמי המתאים לתמונה היעד: חשב את הוקטור הסמי הממוצע ( $\text{avg}_w$ ) ואת סטיית התקן שלו בהתבסס על דוגמאות סמיות אקראיות.

- טען את מחלץ התוכנות של VGG16, אשר ישמש לחישוב perceptual loss על ידי השוואת תכונות level high של תמונות שנוצרו על ידי הגנרטור ושל תמונת היעד.

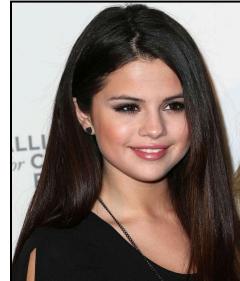
5. לולאת אופטימיזציה: - בצע אופטימיזציה סמייה באמצעות backpropagation המשמש ב- Gradient Descent. בכל שלב:

- צור תמונה חדשה באמצעות הוקטור הסמי הנוכחי והגנרטור StyleGAN2.

- תבצע degradation על התמונה שנותרה כפי שנעשה על תמונה היעד.
  - חשב את ה loss perceptual בין התמונה שנותרה לתמונה היעד.
  - backpropagate את ה loss ועדכן את הווקטור הסטטי בכיון שמקtin את ה loss.
  - תבצע רגולריזציה על המרחב הסטטי על ידי ענישה על סטיות מהווקטור הסטטי המומוצע.
7. סיום: נחזיר את הווקטור הסטטי המקורי וניצור את התמונה המשוחזרת.



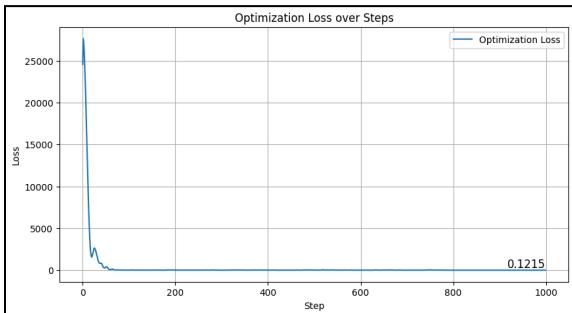
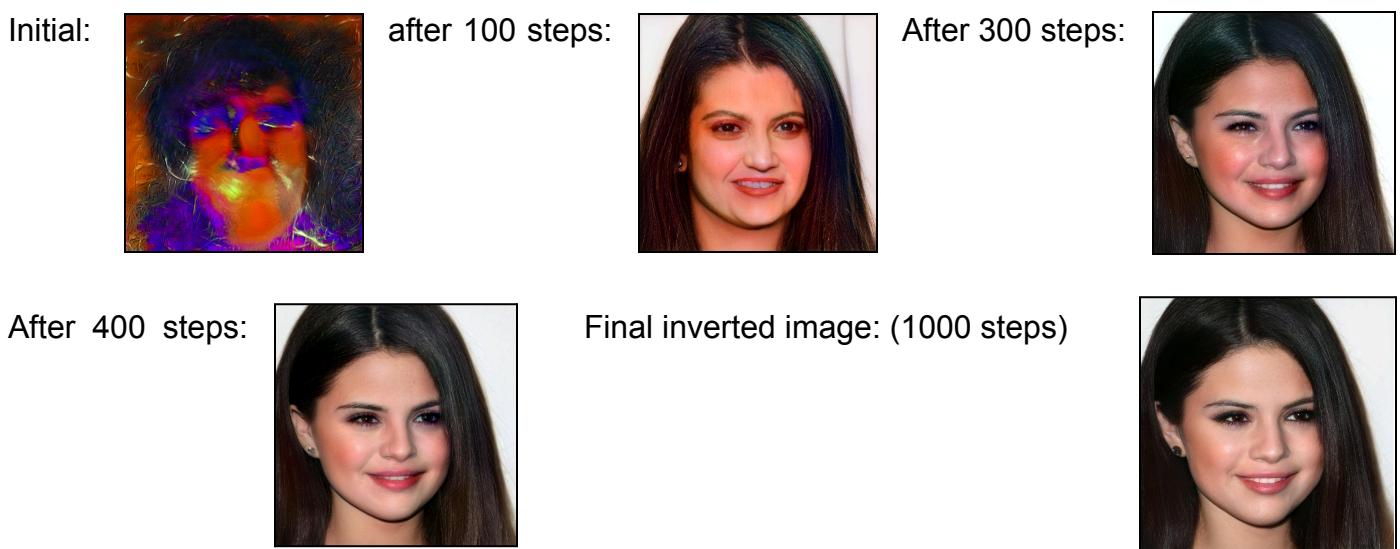
אחרי היישור:



### 3.1 תוצאות

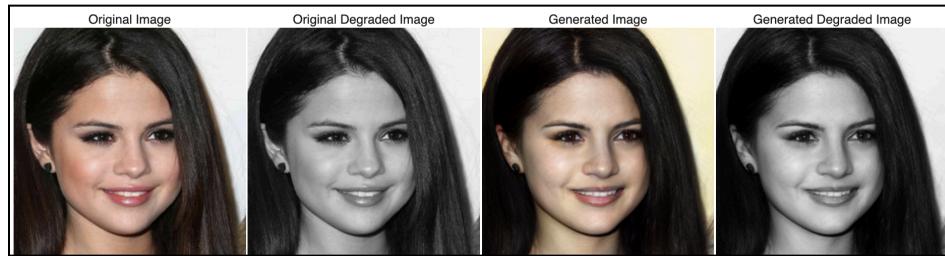
לפני היישור: (התמונה המקורי)

`num_steps=500, latent_dist_reg_weight=0.001 (default values)`



כדי לשמר על ניקיון ובחריות הגרף ולהציג את המגמה הכוללת ללא רעשים, הוצגו רק דגימות של ערכי ה loss. ניתן לראות שלאחר בערך 150 איטרציות ה loss כבר נהיה ממש קטן, אך הינו adaptation קטנים עדין משמעותיים, כמו שנitin לראות את השיפור מהתמונה באיטרציה ה 400 לאיטרציה الأخيرة (1000). ניתן להסביר זאת מכך שבני אדם רגשים יחסית לתמונות פנים, זה משחו שאנחנו רואים באופן תמידי מאד וההבדלים בין אנשים שונים לעתים יכולים להיות קטנים, כך למשל לעתים ניתן ללמוד להבחין אפילו בין תאים זחים שונים בנקודות חן וכדומה. על כן המוח

שלנו הסתגל לזרות דקיות קטנות והשיפור הקטן ב loss משמעוší בرمת תמונה התוצאה עבר בנו אדם.



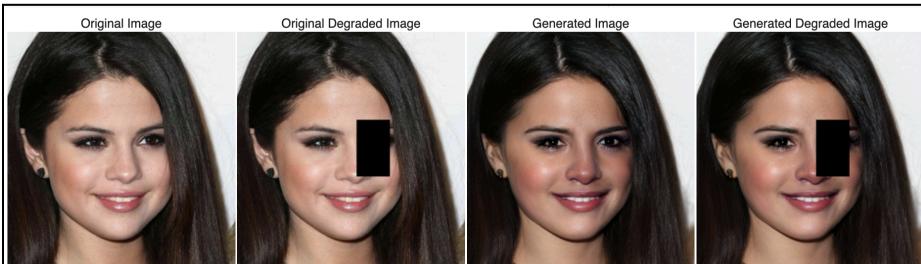
### 3.3.1 תוצאות

הפרמטרים בהם השתמשתי:

`num_steps=1000`  
`.latent_dist_reg_weight=0.5`

כדי לשחרר את הצבע, יצרתי

פונקציית דגרדציה ש יודעת לקבל tensor ארבע מימדי ([W, H, C, B]) שמחזיק תמונות עם שלושה ערוצי צבע b,g,r (בעיבוד המקיים אנחנו הופכים את תמונה הקלט לגנרטור לתמונות rgb ולכן זה יעבד) ולפי מטריצת המעבר מ RGB ל IQY שלמדו בשיעור הראשון, העברתי מ RGB color מ representation luminance, grayscale representation (לפי השורה במטריצה המתאימה ל representation ל luminance component). מכיוון שהקוד מצפה לתמונה המוצפה עם 3 ערוצי צבע כפי שצווינן בהוראות התרגום (כדי שהפיצרים יחולצו מהתמונה על ידי המחלץ באופן תקין), נscalפ את ערכי האפור לשולשת ערכי הצבע, כך שלכל הערוצים יש אותו ערך כמו בתמונה בגווני אפור. קראתי לפונקציה הזאת apply\_grayscale\_degradation, והיא נקראת מתוך `vert_image_invert` כפי שצווין באלאgorיתם, כדי להcin את התמונה במצב שהוא צריך. זהה למעשה התמונה שנשלחה לפונקציה run\_latent\_optimization, בה מנוטים לעשות שחזור. בכל איטרציה של הפונקציה run\_latent\_optimization, יש קראיה נוספת לפונקציית הדגרדציה הניל' בשביל חישוב ה loss, כך שהתמונה שניצור מהגנרטור תתקרב ככל הניתן לתמונה המקורית (לפני הדגרדציה). מכיוון שככל מה שיש לנו זאת התמונה ה degraded, נרצה להפעיל על פלט הגנרטור את פונקציית הדגרדציה, כי כמובן שלא נרצה להשווות בין תמונה צבעונית (הפלט של הגנרטור) לתמונה בגווני אפור (התמונה המקורית שעברה דגרדציה, שהיא כל מה שיש לנו בפונקציה `run_latent_optimization`). נשים לב שמכיוון שהגנרטור הוא בעל ידע מוקדים על תמונות פנים, וכן יש לו תנאי רגולרייזציה ששומר אותו יחסית קרוב לתמונה מציאותית וטבויות בDATA סט עליון אומן, על אף שהגנרטור לעולם לא ראה את התמונה המקורית, ואין לו מידע על ערוצי הצבע, אלא רק על מידת ההארה (למשל אין לו את ערוצי q, בשביל שחזור מדויק), סביר שלא נקבל שחזור מדויק, כפי שניתן לראות בתמונה התוצאה, אך כן נצליח לקבל צבעים מתאימים לתמונה פנים אנושית (שפתיים ורודות, גוון עיר בפרצוף) המזכירים ונראים טוב ביחס לתמונה המקורי. הקושי שנטקלתי בו הוא שנדרשו מספר הרוצות (כאשר הייתה מגבלת GPU) עם פרמטרים שונים כדי לנסות לקבל תמונה תוצאה אופטימלית. בהחלטת יתכן שאם לא הייתה מגבלה זאת, הייתה בודקת פרמטרים נוספים, הייתה מתקבלת תוצאה טוביה יותר. בהתחשב המגבלה הקיימת, השתדלתי לספק תוצאה הכי טוביה שיכולה.



### 3.3.3 תוצאות

הפרמטרים בהם השתמשתי:

`num_steps=1000,`  
`.latent_dist_reg_weight=0.7`

הוסף פונקציה שמיישמת דגרדציה באמצעות יצירת מסכה על תמונות. הפונקציה בשם `apply_inpainting_degradation`, מקבלת טנזור ארבע מימדי (`[W, H, C]`) המכיל תמונות צבע מנורמלת לטווח [0,1] בפורמט RGB. הפונקציה יוצרת מסכה המאפשרת 'הסרת' חלק מהתמונה על ידי הצבת ערך אפס (שחור) באזורי מסוימים לבחירה, על מנת לדמות סיטואציה שבה חלק מהתמונה אינם נראה או דרוש שחזור. בדומה לשעיף הקודם, ישנה קריאה בפונקציית `shoutvert_image` לפונקציית הדגרדציה (כאשר הארגומנט של הדגרדציה מתאים) כדי שהתמונה `shoutvert_latent_optimization` יקבל תהיה ללא המידע המקורי על האזור, בניסיון לשחזרו. כל איטרציה בפונקציה `shout_latent_optimization` מנסה להתקרב ככל האפשר לתמונה המקורי לפני הדגרדציה, על ידי הפעלת הדגרדציה על תמונה האופטימופט של הגנרטור עם הוקטור הסמי העכשווי, וחישוב ה `loss perceptual` על ידי הפעלת מחלץ התמונות על תוצאה זאת. כך, הגנרטור 'לומד' למלא את החסר בתמונה על מנת לשחזר אותה بصورة הטובה ביותר. נשים לב שהשחזר אף פעם לא יהיה מושלם, מכיוון שהאזור המקורי המכוסה אף פעם לא נראה על ידי הגנרטור. מכיוון שיש לו `prior knowledge` על פנים, אז הוא כן מצליח להשלים את האזור החסר بصورة סבירה שמייצרת המשויות הגיונית וונונת מראה של פנים אנושיות, אבל כמובן שלא נצליח לעולם לקבל שחזר מדויק של החלק הזה. עדין, ביחס לעובדה שהגנרטור לעולם לא ראה את האזור החסר בתמונה המקורי, הוא מצליח ליצור תמונה מציאותית הדומה למקור באופן מרשימים כפי שניתן לראות בתמונה התוצאה. בדומה לשעיף הקודם, האתגר בתחום זה היה להציג לתמונה האופטימלית תוך ניסוי ושגיאה של פרמטרים שונים, תוך כדי מגבלות טכניות של GPU וזמן הריצה. בנוסף, נתקלתי באתגר שהדגרדציה בפונקציה `shout_latent_optimization` יוצרת מסכה אפורה ולא שחורה. כדי להתמודד עם זה ביצעתנו נרמול של התמונה שנשלחה כארוגמנט לפונקציית הדגרדציה, וביצעתנו דנורמליזציה על תמונה החזרה.

## סיכום

הפיתוח של טכניקה לשחזר תמונה פנים באמצעות גנרטור StyleGAN2 הוא גישה חדשה המאפשרת שחזר מרשימים של תמונות שעברו דגרדציות שונות. היכולת לקודד תמונה דגרידית לתוך מרכיב סמי של GAN ולאחר מכן לשחרר מחדש מחדש, מראה על פוטנציאלי רב של למידת מכונה בהבנת תמונות ובפרט בתפיסת מאפייני פנים אנושיות. על אף שנעשה שימוש באופטימיזציה סמיות-Gradient-Descent כדי למצוא את הוקטור הסמי המתאים ביותר, בתחום השחזר מושפע במידה רבה מהידע המקיים של הגנרטור על תמונות פנים. ככלים כמו `loss perceptual` מסייעים לשפר את דיוק השחזר על ידי השוואת תכונות מפתח בין התמונה המקורית לתמונה שנוצרה על ידי הגנרטור, וכן מספקים מدد לאיכות השחזר. התמודדות עם מגבלות החומרה ומשאבי החישוב, כגון מגבלות GPU, היא חלק מהאתגר ומשפיעה על הסיכוי למצוא את הוקטור הסמי האופטימלי בזמן סביר. תוצאות השחזר מראות על יכולת מרשימה של המודל ליצור תמונות ריאליות הדומות למקור, אף על פי שלרוב לא מצליח לשחזר מושלים (למשל באזורים שלא נראה על ידי הגנרטור). התהילה האיטרטיבי של האופטימיזציה ובחינת פרמטרים שונים מדגיש את החישיבות של ניסוי וטעייה ואת הצורך בזמן ובמשאבים לקבלת תוצאות מיטביות. נדרש כוונון היפר פרמטרים של `latent dist reg weight`, `and steps` ושהן, כדי לקבל tradeoff אופטימלי בין תמונה מציאותית ליכולת ללכת אחרי הדפוסים שבתמונה, וכן בין זמן התכנסות לתוצאה מיטבית, אשר כפי שצוין בריפורט, יכול היה להיות אופטימלי יותר במידה ולא היו משאבים מוגבלים של GPU.