**האוניברסיטה הפתוחה**

**המחלקה למתמטיקה ולמדעי המחשב**

**שיפור רזולוציה של תמונות בעזרת רשתות GAN**

פרויקט גמר זה הוגש כחלק מהדרישות לקבלת תואר

"מוסמך למדעים" M.Sc. במדעי המחשב

באוניברסיטה הפתוחה

המחלקה למתמטיקה ולמדעי המחשב

על-ידי

**ליאור בומוורצל**

הפרויקט הוכן בהדכרתה של ד"ר מיריי אביגל

פברואר 2023

# תקציר

הגדלת רזולוציית התמונה משמעה הפיכת התמונה לחדה וברורה יותר, על ידי הוספת פיקסלים נוספים לתמונה. בפרויקטים כאלה, אלגוריתמים המבוססים על רשתות יריבות זוכים לפופולריות רבה כיוון שהם מייצרים תמונות שנראות אמיתיות וחדות, ויש להן יכולת השלמת טקסטורות הנלמדות מבסיס הנתונים. בשלב האימון, האלגוריתמים מתאמנים על מאגרים רבים של תמונות ברזולוציה גבוהה ולומדים את התכונות והטקסטורות השוליות המאפיינות את העולם האמיתי. במהלך שלב ההפעלה, האלגוריתמים משחזרים את הטקסטורות הללו ומשלימים את המידע הנדרש בהתאם לדרישות המשתמש. בעקבות השלמת הטקסטורות במקומות בהם חסר מידע בתומנה, נוצרת תמונה ברזולוציה גבוהה יותר ובאיכות טובה יותר. אלגורתמים המשתמשים בארכיטקטורה של רשתות יריבות, מייצרים תמונות איכותיות אך לא בהכרח התמונות הכי קרובות לתמונות המקור שרוצים לשחזר, הסיבה לכך היא שהאלגוריתמים מתמקדים בשיפור איכות התמונה על חשבון דיוק השחזור.

תוכן עניינים

[תקציר 2](#_Toc145077134)

[1.מבוא ...................................................................................................................................4](#_Toc145077135)

[2. סביבת העבודה .....................................................................................................................5](#_Toc145077136)

[3.ייצור בסיס הנתונים 5](#_Toc145077137)

[4.מבנה הפרויקט............... 6](#_Toc145077138)

[5.אופן המדידה והשוואה בין הארכיטקטורות............. 6](#_Toc145077139)

[6.מתווה ניסויים .....................................................................................................................8](#_Toc145077140)

[7. SRRESNET 9](#_Toc145077141)

[8. SrGan 12](#_Toc145077142)

[9.בעיית קריסת מצב .................................................................................................................14](#_Toc145077143)

[10 WGAN 17](#_Toc145077144)

[11. WGAN-GP 20](#_Toc145077145)

[12. Pix2Pix 22](#_Toc145077146)

[13. SinGan 25](#_Toc145077147)

[14.מציאת התמונות עם השיפור הגדול ביותר בין הארכיטקטורות. 26](#_Toc145077148)

[15.הצגת התפלגות ווקטור המפאיינים ברשת inception 27](#_Toc145077149)

[16.סיכום.............. 29](#_Toc145077150)

[17.רשימת מקורות 30](#_Toc145077151)

## מבוא

הגדלת רזולוציה בתמונות דורשת לייצר מידע אשר לפעמים לא ניתן לשחזר מהתמונה הקטנה מכיוון שמידע מתמונה מוגדלת יכול להיות מוצג ממספר גדול של פיקסלים ובתמונה המוקטנת יוצג כפיקסלים בודדים. הטקסטורה אשר אוחדה לכמה פיקסלים בודדים, צריכה להתרחב למספר פיקסלים יותר גדול, ועם יותר הפרדה.

בעבודה זו, אבצע שימוש באימון רשתות יריבות בארכיטקטורת Adversarial Generative Networks [1] )בראשי תיבות (GAN לצורך הגדלת רזולוציית תמונה. ארכיטקטורות GAN מאפשרות למידה של התפלגות המידע מהעולם האמיתי ושימוש בו בשעת יצירת התמונה המוגדלת בהשלמת המידע החסר.

מאז מאמר [1] בו ארכיטקטורת GAN הודגמו לראשונה, התפרסמו עוד מגוון רחב של ארכיטקטורות אשר בכולן העקרון הבסיסי, שימוש ברשתות יריבות המתחרות ביניהן. מטרת האחת (הרשת הגנרטיבית) הוא ליצור תמונה, ומטרת השנייה (הרשת הדיסקרימנטיבית) לסווג האם התמונה באה מהעולם האמיתי או יוצרה על ידי הרשת הגנרטיבית.

לצורך מימוש הגדלת רזולוציה, בעבודה זו אתבסס ראשית על מאמר [2] אשר הציע את ארכיטקטורתSRGAN , אותה נתאר בהרחבה בהמשך. המאמר הראה הגדלת רזולוציה עד פי 4 מתמונת מקור מוקטנת. לאחר שאראה את יכולות הארכיטקטורה הזאת על מספר דוגמאות, אשתמש בעקרונות אשר הוצעו במאמרים הנוספים לארכיטקטורות GAN לצורך בחינה האם הן יכולות להוות שיפור לארכיטקטורת SRGAN.

לצורך השוואה בין תוצרי הארכיטקטורות אשתמש בכמה מדדים להשוואה בין תמונות, אשר ארחיב עליהם בהמשך. המדדים יבדקו את התוצרים בכמה מובנים של מרחק בין תמונת מקור לתמונה המיוצרת, כגון מרחק פיקסלים בין תמונת מקור לתמונת מוצא, או השוואה של המאפיינים והסטטיסטיקות של מקבצי תוצרי הרשת הגנרטיבית למול מקבצי תמונות אמיתיות.

עבודה זו היא המשך של עבודה מסכמת אשר ביצעתי בנושא שיפור איכות תמונה בעזרת רשתות GAN. לצורך השלמת הסברים אפנה לעבודה המסכמת.

## סביבת העבודה

הפרויקט נכתב בשפת Python3 בעזרת שימוש ב TensorFlow גרסה 2.10.  
 TensorFlow היא ספריית קוד פתוח ללמידת מכונה, המפותחת על ידי חברת גוגל לבנייה ואימון רשתות עצביות.

ספריית TensorFlow נותנת פתרונות ותשתית לאימון מודלים ללמידת מכונה אשר בהם השתמשתי:

* TensorBoard משמש כסט כלים לוויזואליזציה של TensorFlow. הוא מספק ממשק משתמש ידידותי בו ניתן לבחון ולפרש את הריצות והגרפים של התקדמות האימון לגרפים השונים.
* Keras הוא API של רשתות עמוקות, כתוב ב-Python ומסוגל לרוץ על TensorFlow כ-API ידידותי למשתמש. Keras מקל על בנייה ואימון מודלים של למידה עמוקה. הוא מספק פונקציונליות להגדרה ואימון מודלים של רשתות עמוקות .Keras גם תומך במגוון רחב של סוגי שכבות, פונקציות הפסד, אופטימיזטורים, ועוד.
* ניתן לטעון נתונים ולעבדם מראש באופן יעיל באמצעות API tf.data, שיכול להתמודד עם כמויות גדולות של נתונים, לבצע המרות מורכבות ולטעון נתונים באופן מקביל ויעיל.

החומרה עליה הורצו המודלים והבדיקות :

מחשב 9500 XPS Dell

מעבד Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz

זיכרון RAM 32.0 GB (31.8 GB usable)

GPU NVIDIA GeForce GTX 1650 Ti

## ייצור בסיס הנתונים

בעבודה זו השתמשתי במאגר המידע CelebFaces Attributes (CelebA) Dataset, מאגר מידע מזה מכיל

מקבץ של כ 202.5 אלף תמונות פרצופי מפורסמים, התמונות מיושרות וחתוכות כך שדמות המפורסם במרכז התמונה. גודלי התמונות המקוריות הן 218x178.

לטעינת התמונות ייצרתי מחלקה עוטפת הנקראת MyDataLoader הנמצאת בקובץ DataLoader. המחלקה מייצרת מקבצים (באטצ'ים) של טנסורים בעזרת שימוש בתשתית הניתנת על ידי ה tf.data API של TensorFlow. המחלקה מקבלת כפרמטר את גדלי המקבצים (batch size), גודל התמונה הגדולה, יחס ההקטנה, כמות התמונות אשר יישמרו למקבץ המבחן .validation תמונות ממקבץ ה validation לא יוצגו למודלים בשלב האימון, תמונות אלה ישמשו לבדיקת איכות תוצרי הארכיטקטורות השונות.

בעבודה זו השתמשתי בפרמטרים לאימון הדומים לפרמטרים לאימון מהמאמר [2]. מאמר [2] מקבל מקבצים של זוגות של תמונות, כאשר גודל התמונה ברזולוציה הגבוהה 96, יחס ההקטנה לרזולוציה הנמוכה 4. השארתי 5% ממקבץ התמנות ל validation, זוהי כמות גבוהה יחסית של מקבץ validation. הסיבה לגודל הנבחר בסט הוולידציה היא שביצעתי שימוש בתמונות הוולידציה במדידת מטריקות למרחקים בין מקבצי תמונה אשר דורש כמות גדולה יחסית של תמונות לצורך קבלת תוצאה מדויקת יותר. בשלב טעינת התמונות אני מבצע צימוד של כל תמונה ברזולוציה גבוהה לתמונה אשר עוברת הקטנת bicubic, כך שכל איבר ממאגר הנתונים שאני יוצר מורכב מ-2 תמונות - אחת ברזולוציה גבוהה והשנייה ברזולוציה נמוכה.

## מבנה הפרויקט.

בפרויקט זה בחנתי מספר ארכיטקטורות של רשתות GAN אשר יבצעו סופר רזלוציה על אותו מקבץ תמונות ועם אותם מבחני איכות עבור התוצאות. לכן בפרויקט זה, בחרתי להשתמש בעקרון של פולימורפיזם ולממש מחלקה אבסרטקטית כך שהקוד עבור הפעולות המשותפות ייכתב פעם אחת. מהמחלקה האבסטרקטית לא ניתן לייצר אובייקטים. תפקיד המחלקה האבסטרקטית הוא להוות בסיס לקוד משותף עבור המחלקות הנוספות אשר יורשות ממחלקה זו וממשות את הפוקציות אבסטרקטיות, הפונקציות האבסטקרטיות הן פונקציות המגדירות את צעדי האימון לגנרטור ולדסקרימנטור. כל שאר הפונקציות מלבד פונקציות צעדי האימון כגון, לולאת האימון, מדידות הביצועים, שמירת תוצרי הביניים וחישוב המדדים לאורך התקדמות האימון נעשים במחלקה האבסטרקטית. בעקבות מימוש המחלקה האבסטרקטית, עבור כל מודל שאבדוק בפרויקט, הדבר היחיד שצריך לעשות על מנת להעריך את טיבו הוא לרשת מהמחלקה האבסטרקטית ולממש את צעדי האימון לגנרטור והדיסקרימנטור. הארכיטקטורות השונות אשר אני בוחן, עושות את השינוי הנדרש בקבצים שונים אשר שמם נקבע לפי שם הארכיטקטורה שהם ממשים. המחלקה האבסטרקטית נמצאת בקובץ train.py ונקראת AbsTrainer.

לשם הפרדה בין צורת האימון ליצירת המודל, בחרתי להפריד את יצירת המודלים לקובץ אשר מיועד לכך ונקרא model.py. בקובץ זה, אני בונה את כל המודלים השונים המתאימים לארכיטקטורות השונות.

מחלקה display\_handler מממשת את הצגת התמונות ושמירתם, המחלקה מממשת הצגה של גרפים לפיצרים של תוצרי ארכיטקטורות על מקבצי תמונות. בנוסף מחלקה זו מוצאת ומציגה את התמונות אשר עליהם השינוי גדול ביותר בין תוצרי הארכיטקטורות השונות.

בקובץ run.py אני טוען את התמונות לאימון ולבחינה, מייצר את האובייקטים המממשים את המחלקות השונות ומריץ את האימון שלהם אשר מייצר את התוצרים של הרשת הגנרטיבית, מודד את ערכי המדדים השונים, ושומר את המשקולות לאורך התקדמות האימון.

בקובץ model.py מיוצרים המודלים השונים, המודלים השונים הם הדיסקרימנטורים והגנרטור הרולוונטיים לכל מחלקה. הפונקציות שמייצרות את המודלים מקבלות פרמרטים אשר מאפשרים גמישות ביצרית המודלים, שינויים שניתן לבצע במודלים הם בחירת כמות השכבות, וגדלי הפילטרים.

## אופן המדידה והשוואה בין הארכיטקטורות.

אלגוריתמים המשתמשים בארכיטקטורה של רשתות יריבות, מייצרים תמונות איכותיות אך לא בהכרח התמונות הכי קרובות לתמונות המקור שרוצים לשחזר. הסיבה לכך היא שהאלגוריתמים מתמקדים בשיפור איכות התמונה על חשבון דיוק השחזור. לצורך שיפור איכות התמונה האלגוריתמים מוסיפים מידע וטקסטורות ,המידע שיושלם על ידי הגנרטור יהיה כזה שיגרום לדסקרימנטור לסווג בצורה שגויה להאם אוביקט הגיע מהעולם האמיתי.

אין הגדרה מסויימת איך התמונות הסופיות בתוצרי הרשת אמורים להראות. האם ייצור תמונה עם מאפיינים חדים אך שונים במיקצת מהמקור עדיף על תמונה מטושטשת במיקצת אבל עם מאפיינים דומים למקור?

השוואה בין ארכיטקטורת לרשתות יריבות זו משימה קשה. למשל כל ארכיטקטורה יצרה מאפיינים משלה, איך משווים בין מאפיינים שונים (למשל פיזור שער ראש אשר נראה אמיתי אבל שונה בכל תוצר ארכיטקטורה) ? או לאיזה מדד נותנים עדיפות בציון, לחדות או לדיוק?

בעבודה המסכמת בפרק 5 תיארתי בהרחבה כמה מדדים להשוואה בין תוצרי הרשתות ומה היתרון של כל אחד מהם. בעבודה זו התמקדתי ב 3 מדדים אשר משווים בין הארכיטקטורות בשתי מובנים, מדדי PSNR ו SSIM אשר משווים כמה תמונה אחת קרובה לשניה מבחינת מרחק פיקסלים. ומדד FID אשר משווה כמה מאפיינים בין מקבצי תמונות דומים.

המדדים מוסברים בהרחבה בעבודה המסכמת. אסביר בקצרה על המדדים אשר השתמשתי בעבודה זו:

Peak signal-to-noise ratio PSNR – מודד את היחס המרבי בין האות לבין הרעש, הוא נמדד בדציבלים. בהינתן תמונה I ותוצר של תמונה משופרת K ניתן למדוד את ההבדל בין התמונות כמקסימום הערך בתמונה חלקי הפרש הערך הריבועי של כל פיקסל בין התמונות. החיסרון במדד זה הוא בכך שהוא לא מעניש עבור שינויים המושפעייים שונה עבור התפישה האנושית. למשל עבור דוגמא בה התמונה הוזזה בפיקסל אחד נקבל ערכים נמוכים, למרות שבעין האנושית לא נוכל להבחין בהבדל, זאת מכיוון שכל פיקסל עכשיו יושווה לפיקסל שלא מתאים לו מבחינת מיקום. בעבודה עשיתי שימוש במימוש של ספריית TF למדד זה.

Structural similarity SSIM - [ [5הוא מדד מרחק של ערכים סטטיסטיים הבוחן אזורים בתמונה. המדד מנסה להעריך מרחק בין שתי תמונות לפי התפיסה של מבנה התמונה כפי שמערכת הראיה האנושית מסתכלת על תמונה. המדד פועל על ידי חלוקה לאזורים והשוואה של מבני האובייקטים בתמונה התאורה, והניגודיות. המדד סימטרי ותוצאות המדד הם בין -1 ל 1, כאשר ככל שהתמונות דומות יתקבלו ערכים קרובים ל-1. בעבודה עשיתי שימוש במימוש של ספריית TF למדד זה.

Fréchet Inception Distance FID - [6] במדד זה משווים את התפלגויות המאפיינים של האובייקטים הנוצרים בתמונת מהרשת הגנרטיבית להתפלגויות של המאפיינים של האובייקטים בתמונות מהעולם האמיתי. ככל שהתפלגויות האיברים המיוצרים על ידי הרשת הגנרטיבית שונות מהתפלגויות האיברים בעולם האמיתי, נקבל כי המרחק יגדל. לחישוב המאפיינים של האובייקטים בתמונות משתמשים ברשת קלסיפיקציה נוספת אשר אומנה על משימת סיווג תמונות לפני השימוש בה בחישוב במדד. מעבירים מקבצי תמונות מהעולם האמיתי ומקבצי תמונות מתוצרי הרשת הגנרטיבית ברשת הנוספת, ומשתמשים בתוצאות ערכי השכבה הפנימית לחישוב ווקטורי המאפיינים של האוביקטים בתמונה. על ווקטורי המאפיינים הנוצרים מבצעים חישוב סטטיסטי של מרחק בין התפלגויות (מדמים את ההתפלגויות לנורמליות לנוחות החישוב אשר משקלל את המרחק של ערכי הממוצע וערכי השונות המשותפת:

– שונות

- ממוצע

בפרויקט זה מימשתי את מדד FID לצורך מדידת מרחקים בין מקבצים של תמונות. אני משתמש במדד זה בשלב מדידת התקדמות הארכיטקטורה. המימוש משתמש ברשת inception מאומנת אשר הורדה ממנה השכבה העליונה, לצורך הוצאת ווקטורי המאפיינים של התמונות. לאחר הוצאת ווקטורי המאפיינים מבוצע החישוב לקבלת המרחק בין המקבצים כפי שמופיע בנוסחה למעלה.

לצורך מדידה של התקדמות האריכטקטורות או השוואה ביניהם כתבתי את הפונקציה -

calculate\_fid\_ssim\_psnr\_cur\_step .

בכל המדדים אשר הזכרתי למעלה מושווים תוצרי ארכיטקטורת הגנרטור לתוצרים מהעולם האמיתי.

על מנת לחסוך זמן חישוב, כתבתי את הפונקציה כך שהיא משתמשת באותה קריאה לגנרטור לייצור התמונת עבור חישוב מגוון המדדים.

## מתווה ניסויים

בעבודה זו אני מבצע מספר ניסויים, מטרתם של הניסויים הם לבחון שינויים בארכיטקטורת SrGan, לצורך שיפור יציבות האימון ואיכות התוצאות הסופיות. השינויים הנבחנים הם שיפורים לארכיטקטורת GAN אשר מסוקרים בעבודה המסכמת שביצעתי. שינויי הארכיטקטורה מתוארים בפרקים הבאים במסמך זה.

הניסויים שביצעתי לפי פרקים:

* פרק 7 – בחינת ארכיטקטורת SrResnet אשר מנסה למזער מרחק פיקסלים בין תמונות מקור לתמונות מיוצרות.
* פרק 8 – בחינת ארכיטקטורה SrGan אשר משתמשת באותו מודל מSrResnet ומשנה את צורת האימון. צורת האימון היא רשתות יריבות ופונקציית הפסד שמנסה למזער מרחק פיצרים. בפרק זה אני משווה בין SrGan ל SrResnet. התוצאות והמסכנות אשר אותם אני מקבל דומות לתוצאות המתקבלות ממאמר [2].
* פרק 9 - ביצוע אימון SrGan ללא transfer learning . בניגוד למאמר [2], אשר מתארים שימוש במשקולות מאותחלות לאחר transfer learning מארכיטקטורת SrResnet . בפרק זה ניסיתי לבצע אימון ממשקולות אקראיים.
* פרק 10 – החלפת פונקציית ההפסד למרחק ווסרשטיין. מטרת שינוי זה היא לשפר את יציבות האימון ולהפחית את קריסות המצב.
* פרק 11 – שינויי תחימת המשקולות בתנאי רגולציה השומר על תנאי ליפשיץ שמחוייב במרחק ווסרשטיין. מטרת שינוי זה היא להקל על בחירת הפרטמטר לתחימת המשקולות.
* פרק 12- בחינת מספר שינויים המוצאים בארכיטקטורת pix2pix , בפרט בחנתי שימוש בדיסקרימנטור PatchGan, בחלק זה בעבודה בחנתי מספר משקלים שונים בין הגורמים בפונקציית ההפסד כפי שמופיע במאמר [8], התוצאות שקיבלתי עבור המשקלים השונים בין הגורמים בפונקציית ההפסד מתאימים לתוצואת ממאמר [8].
* פרק 13 – בחינת ארכיטקטורה אשר מבצעת progressive growing שאמור לעזור בייציבות האימון ,הארכיטקטורה שאני בוחן בפרק זה מבצעת שימוש במרחק ווסרשטיין עם תנאי רגולציה על תנאי ליפשיץ כפי שמבוצע בפרק 11. בפרק זה בחנתי איך עוזר הקפאת המשקולות ב progressive growing כאשר ביצעתי אימון ללא הקפאה ואם הקפאה.

## SRRESNET

מבנה רשת הגנרטור במאמר [2] בנויה בצורת חיבורי blocks residual ו skip connections. חיבור residual מתואר במאמר [4], הוא חיבור בין כניסת שכבה (או בלוק שכבות) למוצא שלה. חיבור skip connection מתאר חיבור בין שכבות תחתונות לעליונות.ארכיטקטורה במבנה שכזה מתמודדת טוב בהעלמות הגרדינאט בלמידה הנעשית בעזרת אלגוריתם gradient descent. באלגוריתם gradient descent מנסים לחשב את כיוון שינוי המשקולות ברשת. שינוי המשקולות יהיה בכיוון השיפוע שיקטין את ערך פונקציית ההפסד. חישוב הגרדיאנט בשכבות הראשונות מושפע מגרדינאטים בשכבות הסופיות ולכן ככל שמוסיפים יותר שכבות, כך השינויים בשכבות הסופיות עלול להיות קטן או חסר משמעות לשכבות הראשונות. בחיבור residual ו skip connections מדלגים על חיבור השכבות המרובות, ולכן השינוי בערכי שכבות סופיות משפיע יותר על שכבות ראשונות, וחישוב הגרדיאנט של השכבות הראשונות ביחס לסופיות מושפע בהתאם.

בנוסף**,** חיבור מסוג residual מספק לנו את האפשרות להעביר את פונקציית הזהות בין השכבות אשר הוא מחבר. המשמעות היא שכל שכבה המחוברת באופן זה צריכה להתמקד בלימוד של שינויים בסיסיים בלבד, הנוגעים לתוצאות השכבה הקודמת לה, דבר זה יוצר הקלה ללמידה.

בשל העקרונות המתוארים מעלה אשר מקלים על האימון ועל העלמות הגרדיאנט, ארכיטקטורת resnet מאפשרת להשתמש ביותר שכבות, שימוש ביותר שכבות מאפשר ליצור פונקציות מיפוי יותר מורכבות. בעבודה המסכמת אני מתאר בהרחבה ניסוי שמחברי מאמר [2] מבצעים והוא לבחון את השפעת כמות הבלוקים בחיבור residual. תוצאות הניסוי מראות שככל שמגדילים את כמות הבלוקים מקבלים אימון יותר יציב ותוצאות בעלות מדדי איכות טובים יותר, ומצד שני ככל שהרשת גדלה בעקבות הוספת הבלוקים האימון נעשה איטי יותר.

בפרויקט זה מימשתי את מבנה רשת הגנרטור כפי שמופיע במאמר 2. באיור 1 ניתן לראות את החיבור בין ה blocks residual ו skip connections, כאשר כפי שמוצע במאמר השתמשתי ב 16 blocks residual. בכל בלוק יש חזרה של השכבות בסדר הבא: שכבת קונבולוציה , שכבת באטץ ושכבת prelu כאשר k מתאר את גודל הקרנל, n מתאר את כמות הפילטרים ו s מתאר את גודל הסטרייד.

בשלב הגדלת התמונה (הריבועים ברקע ירוק) משתמשים בשכבת pixel shuffler כפי שהוצע במאמר [11], שכבה זו מבצעת הגדלת תמונה בעזרת פרמטרים נלמדים. השכבה ממירה את האלמנטים בטנזור ממרחב העומק (כמות ה channel) למרחב גודל התמונה (גובה ורוחב). כאשר השתמשתי בשכבה זו נוצרו לי ארטיפקטים בתמונות הסופיות בצורת ריבועים. זוהי בעיה מוכרת אשר הפתרון לה הוא שינוי אופן הגדלת התמונה. בפרויקט החלפתי את בלוקי הגדלת התמונה בבלוקים שראשית מבצעים הגדלת תמונה בפרמטרים לא נלמדים על ידי שיטת nearest neighbor, אשר אמורה ליצור תמונה מוגדלת ומטושטשת, לאחר שכבת ההגדלה מבצעים שכבת קונבלוציה עם פילטר במשקולות נלמדים לאיך לשנות ולחדד את התמונה המוגדלת, רעיון זה מתואר במאמר [11] גם כן.

A diagram of a block diagram

Description automatically generated with low confidence

איור 1 : מבנה ארכיטקטורה srResNet , מורכב מבלוקים המחוברים בעקרון הresidual ו skip connections . האיור לקוח ממאמר [2]

פונקציית ההפסד בשלב זה באימון היא מרחק MSE בין פיקסלים של התמונה המקורית לתמונה הנוצרת. מרחק זה עוזר לקרב את תמונת המוצא לתמונת המקור ברמת הפיקסלים. במרחק זה מודדים את ההפרש עבור כל פיקסל בנפרד בין תמונת המטרה לתמונת הפלט מהרשת הגנרטיבית. מרחק זה מייצג את המשימה האמיתית עליה אנו נדרשים לענות, והיא לייצר תמונה כמה שיותר דומה לתמונת המקור. אם נחשוב על שחזור תמונת מוצא מושלם שעבר תזוזה של פיקסל אחד בודד, במקרה שכזה המרחק הנמדד יכול להיות גדול, למרות שהתמונה תראה מאוד מדויקת ואמיתית למתבונן האנושי. שימוש במרחק שכזה לרוב יוצר תמונת מטושטשת מכוון שמטרתו להגיע למינימום שגיאה של האפשרויות אשר יכול להיות הממוצע של אותן אפשרויות, ערך שכזה לא בהכרך מייצג ערך אפשרי בעולם האמיתי.

- תמונוה n ברזולוציה גבוהה

- תמונוה n ברזולוציה נמוכה

*- הפעלת הרשת הגנרטיבית על תמונה n ברזולוציה נמוכה*

ביצעתי אימון של ארכיטקטורות SrResNet כפי שתיארתי, כאשר פונקציית ההפסד היא MSE.

באיור 2 ניתן לראות את ערכי פונקציית ההפסד לאורך האימון, קורדינאטות ציר ה X הם כמות צעדי האימון, קורדינטות ציר ה Y הם מרחק הMSE בין הפיקסלים של תמונת המקור לפיקסלים של התמונה הנוצרת מהגנרטור. ניתן לראות כי פונקציית ההפסד לא יורדת באזור של מעל מיליון צעדים, לכן זוהי נקודה טובה לעצור. זוהי הנקודה שבה מחברי מאמר [2] עצרו את אימון SrResNet.  
A picture containing line, plot, font, text

Description automatically generated

איור 2 : גרף המציגים את פונקציית הפסד ארכיטקטורת SrResNet לאורך התקדמות האימון .

בטבלה למטה ניתן לראות את ערכי המדדים לאיכות תוצרי הרשת. מדדי PSNR ו SSIM גדלים ככל שמתקדמים באימון. עליה בערכים אלה משקפת שהקירבה לתמונת המקור גדלה מבחינת מבנה התמונה ומרחק פיקסלים.

ניתן לראות שערכי FID קטנים, ככל שערכים אלה קטנים, מתקבל כי הסטטיטסיקה של ערכי הפיצרים הנוצרים משכבה פנימית של רשת inception מאומנת, עבור מקבצי תמונות מעולם האמיתי ומתמונת מיוצרות על ידי הרשת הגנרטיבית קטנה. נראה כי הגרפים המוצגים מראים האטה במגמות שלהם לקראת סוף האימון ולכן אנו מצפים להאטה במגמת שיפור התוצרים של הארכיטקטורה .

|  |  |  |
| --- | --- | --- |
|  |  |  |

איור 3: גרפים המציגים את מדדי איכות תוצרי רשת הגנרטור כפונקציה של התקדמות האימון עבור ארכיטקטורת SrResNet . משמאל לימין: PSNR, FID, SSIM. קורדינטות ה X מייצגות את מספר צעדי האימון וקורדינטות ה Y מציגות את ערך המדד.

להלן דגימה והצגה של מקבץ תמונות אשר עברו את תהליך ההגדלה, תמונות אלו לא עברו ברשת בשלב האימון, הם נשמרו בצד לצורך הצגה. ניתן לראות בצד שמאל את התמונה המוקטנת, באמצע את תמונת המקור, ובצד ימין את התמונה המשוחזרת. כצפוי התמונות הנוצרות קרובות בבסיסן לתמונת המקור אך מטושטשות. כפי שתואר שימוש במרחק MSE לרוב יוצר תמונת מטושטשת.

A blurry image of a person

Description automatically generated with medium confidenceA person with dark hair

Description automatically generated with low confidence

A person with long hair and beard

Description automatically generated with low confidenceA person with long hair and beard looking to the side

Description automatically generated with low confidence

A person with a beard

Description automatically generated with low confidenceA picture containing human face, person, person, clothing

Description automatically generated

A person smiling at the camera

Description automatically generated with low confidenceA person smiling at the camera

Description automatically generated with low confidence

A close-up of a person

Description automatically generatedA person with blonde hair

Description automatically generated with low confidence

איור 4: הצגת תוצרי ארכיטקטורת SrResnet על תמונת אשר לא בוצע עליהם לימוד. עמודות משמאל לימין, תמונה ברזולוציה נמוכה, תמונה ברזולוציה גבוהה, תמונה הנוצרת מרשת SrResNet

## SrGan

באימון SRGAN משתמשים בTransfer learning לאתחול משקולות הגנרטור. בTransfer learning מתחילים אימון ממשקולות אשר בוצע עליהם אימון עבור משימה קודמת. גם בארכיטקטורת SrGan תחילה מאמנים את הגנרטור ללא הדיסקרימנטור בצורת SRResNet ולאחר מכן ממשיכים את האימון בצורת הרשתות היריבות בצורת SrGan. גם אני כמו במאמר [2], התחלתי את אימון SrGan עם משקולות שנוצרו לאחר אימון SRResNet לאחר מיליון איטרציות אימון. כפי שהמאמר מציע, אימנתי את ארכיטקטורת SrGan באימון של רשתות יריבות עוד 200 אלף צעדים.

בשלב זה במאמר משתמשים בשתי פונקציות הפסד אשר לחיבור ביניהם הם קוראים perceptual loss (מלשון תפיסה).

החלק הראשון הוא שימוש בcontent loss אשר הוצע במאמר [ [3ומוסבר בהמשך. מטרת חלק זה היא לשמור על מבנה התמונה, כלומר לשמור על המאפיינים הבסיסיים מתמונת המקור.

החלק השני בפונקציית ההפסד הוא .adversarial lossבחלק זה נעשה שימוש בערכי תוצאות ההסתברות של הרשת הדיסקרימינטיבית על תמונה שיוצרה מהרשת הגנרטיבית. מטרת חלק זה היא לגרום לארכיטקטורה לייצר טקסטורות אשר דומות לטקסטורות שנדגמו מתמונות אמיתיות אשר איתם אימנו את הארכיטקטורה.

פונקציית ההפסד של אימון הרשתות היריבות על זוגות של תמונות ברזולוציה גבוהה ורזולוציה נמוכה באימון הדיסקרימינטור

פונקציית ההפסד הגנרטור באימון ה adversarialהיא :

פונקציית ההפסד הכללית באימון הגנרטור :

בהצעת המאמר השתמשתי בערך

A picture containing calendar

Description automatically generated

איור 3 : הדגמת איכות התמונה ועדיפות שימוש בSRGAN אשר בעל אותה ארכיטקטורה כמו SRResNet אבל אומן בעזרת אימון רשתות יריבות. התמונה נלקחה ממאמר [2].

להלן הדגמה של תוצרי הרשת מהאימון שלי לארכיטקטורת SrGan.

בתמונת למטה ניתן לראות בעמודות משמאל לימין את התמנות הבאות:

תמונה ברזולוציה נמוכה, התמונה המקורית ברזולוציה גבוהה, התמונה שנוצרה מארכיטקטורת SrResNet, תמונה הנוצא מארכיטקטורת SrGan. ניתן לראות כי התמונות שנוצרו מארכיטקטורת ה SrGan חדות בהשוואה לתמונות שנצורו מארכיטקטורת ה SrResNet.

A blurry image of a person

Description automatically generated with medium confidenceA person with dark hair

Description automatically generated with low confidence

A person with long hair and beard

Description automatically generated with medium confidenceA person with long hair and beard

Description automatically generated with low confidenceA person with long hair and beard looking to the side

Description automatically generated with low confidence

A person with a goatee

Description automatically generated with low confidenceA person with a beard

Description automatically generated with low confidenceA picture containing human face, person, person, clothing

Description automatically generated

A person smiling at the camera

Description automatically generated with low confidenceA person smiling at the camera

Description automatically generated with low confidenceA person smiling at the camera

Description automatically generated with low confidence

A close-up of a person's face

Description automatically generatedA close-up of a person

Description automatically generatedA person with blonde hair

Description automatically generated with low confidence

איור 4 : הדגמת תוצרי הארכיטקטורות משמאל לימין התמונה ברזלוציה הנמוכה, התמונה ברזלוציה הגבוהה , ארכיטקטורת SRResNet , ארכיטקטורת SrGan. ניתן לראות כי תוצרי SrGan חדים יותר מ SrResnet. תמונות אלו נדגמו מסט הוולידציה לאחר השלמת צעדי האימון.

כאשר מתבוננים בערכי פונקציית ההפסד, ניתן לראות שיש ירידה משמעותית יותר בפונקציית הדיסקרימנטור עד לשלב 1.1 מיליון צעדים. נשים לב כי מכיוןן שהדיסקרימנטור הוא מסווג, אשר דרך המשוב שלו, הגנרטור לומד האם התמונות הנוצרות נראות אמיתיות או לא, רצוי כי הדיסקרימנטור לא יאומן עד לאופטימליות שלו. לכן ניתן להבין למה מחברי המאמר אימנו רק עוד 200 אלף צעדים, ולא יותר. בהמשך העבודה אראה אימון ארוך יותר על ארכיטקטורה זו ואת תוצאותיו, בנוסף אסביר את הסיבה למה אימון דסקרימנטור שמגיע לערכי פונקציית הפסד נמוכים מהווה בעיה.

A screenshot of a graph

Description automatically generated with medium confidence

איור 5: ערכי פונקציית ההפסד של הדיסקרימנטור באימון SrGan. אימון בין הצעד המליון לצעד ה1.2 מליון

## בעיית קריסת מצב

קריסת מצב (mode collapse) זהו מקרה בו הארכיטקטורה מייצרת מגוון תמונות מצומצם וממשיכה לאמן ולשפר רק את מגוון התמונות המצומצם הזה.

הסיבה להגעה לקריסת מצב נעוצה באופן עדכון המשקולות של הגנרטור לפי פונקציית ההפסד לארכיטקטורת GAN, עדכון המשקולות נעשה כך שיגרום לרשת הדיסקרימנטור להחזיר הסתברות גבוהה על תוצרי הרשת הגנרטיבית. הערך אשר מוכל בפונקציית ההפסד, הוא החלק היחידי אשר הרשת הגנרטיבית יכולה להשפיע על ערכי פונקציית ההפסד. במקרה שיש אובייקטים עבורם יקבל ערכים גבוהים מאחרים, לרשת הגנרטור יהיה תמריץ להמשיך לייצר אובייקטים אלה. להרחבה בנושא, ניתן לקרוא את סעיף 2.3.3 בעבודה המסכמת.

בשלב זה בעבודה, ניסיתי לבחון האם אכן יש צורך בtransfer learning באיתחול המשוקולות. התחלתי אימון ללא שימוש בשלב המקדים באימון בצורת SrResnetוהגעתי למקרה של קריסת מצב. הפסקתי אימון זה לאחר חצי מיליון צעדים מכיוון שראיתי שהוא לא מתקדם בצורה טובה.

באיור 6 אני מציג בעמודה הימנית את תוצרי הרשת הגנרטיבית עבור מספר תמונות מקור המוצגות בעמודה השמאלית. ניתן לראות איך הרשת מייצרת את אותה תמונת בסיס עבור כל תמונת מקור. ניתן לראות שגוון הרקע נשמר באופן יחסי יותר נאמן לתמונת המקור, ואילו אזור הפנים בתמונה מיוצר באופן דומה יותר בין כל התמונות.

A blurry image of a person's face

Description automatically generatedA person with dark hair

Description automatically generated with low confidence

A blurry image of a person

Description automatically generatedA person with long hair and beard looking to the side

Description automatically generated with low confidence

A blurry image of a person

Description automatically generatedA picture containing human face, person, person, clothing

Description automatically generated

A blurry image of a person

Description automatically generatedA person smiling at the camera

Description automatically generated with low confidence

A blurry image of a person

Description automatically generatedA person with blonde hair

Description automatically generated with low confidence

איור 6: הצגת מקרה של קריסת מצב לאחר אימון SrGan ללא שימוש בtransfer learning , עמודה שמאלית התמונה המקורית, עמודה ימנית התמונה הנוצרת מרשת הגנרטור.

באיור 7, אני מציג את ערכי המדדים לבדיקת איכות. כאשר מסתכלים על ערכי מדד FID ערכו ממש גבוה, ערכו מציג את העובדה שסטטיסטיקת המאפיינים של תמונות מהעולם האמתי לאלה המיוצרות על ידי הגנרטור שונה מאוד. בנוסף ערכי ה SSIM נמוכים דבר המראה כי מבנה התמונה שונה מאוד בין תמונת המקור לתמונה המיוצרת. ערכי ה PSNR באופן יחסי גם נמוכים דבר המראה כי מרחק הפיקסלים בין תמונת המקור לנוצרת על ידי הגנרטור שונה.

A screenshot of a graph

Description automatically generated with medium confidence

איור 7: הצגת ערכי מדדים להשוואה בין ארכיטקטורות לאחר מקרה של קריסת מצב באימון SrGan ללא שימוש בtransfer learning .

באיור 8 אשר מציג את ערכי פונקציות ההפסד עבור הדיסקרימינטור לאורך התקדמות האימון. קורדינטות X מייצגות את צעדי האימון וקרודינטות Y מייצגות את ערכי פוקנציית ההפסד. באיור ניתן לראות שערכי פונקציית ההפסד של הדיסקרימינטור מתאפסים בצורה מהירה. במקרה שכזה לא ניתן משוב יעיל על ידי הדיסקרימנטור לרשת הגנרטור. הרשת הדסקרימינטיבית למדה לפסול תמונות מהסוג הנוצר בקריסת מצב, לכן ערכי פונקציית ההפסד שלה מתאפסים. במקרה זה רשת הגנרטור לא יכולה להתקדם באימון מכיוון שהמשוב של הדיסקרימנטור עבור כל תמונה הוא אותו ערך.

A screenshot of a graph

Description automatically generated with low confidence

איור 8: הצגת ערכי פונקציות ההפסד של הדסקרימנור בשלב האיומן SrGan ללא שימוש בtransfer learning .

באיור 9 לצורך הדגמה של כמות הריכוזיות בקריסת מצב, אני מציג את הגרפים הבאים לפי סדר הגרפים המוצגים משמאל לימין, גרף פיזור האובייקטים ממקבץ תמונת המקור, גרף פיזור האובייקטים ממקבץ של ממקרה קריסת המצב וגרף מפיזור האובייקטים עבור מקבץ מתוצרי ארכיטקטורת SrPix2Pix שאיתה אימנתי ואתאר בהמשך. פיזור האויבייקטים מוצג על ידי ווקטור המאפיינים אשר מחושב ברשת הinception, וקטור המאפיינים הוא ממרחב רב מימדי, על מנת להציג ווקטורים רב מימדיים במרחב דו מימדי אני מבצע אלגוריתם PCA לקבלת מטריצות להמרה עבור הווקטורים המקוריים כך שהוקטוריים החדשים ייצגו את מקבץ המאפיינים עם השונות המקסימילת. הגרפים המוצגים למטה הם גרפים המייצגים צפיפות אובייקטים אשר מוצגים על ידי שתי הווקטרוים עם השונות המקסימלית.

A picture containing screenshot, colorfulness, graphics

Description automatically generatedA picture containing screenshot, colorfulness, graphics

Description automatically generatedנשים לב שהסקאלה של הצפיפות לא מנורמלת בין הגרפים, והסקאלה של מקרה קריסת המצב הוא בפקטור 10 יותר גדול, כלומר המאפיינים הרבה יותר מרוכזים באותה נקודה.

A picture containing screenshot, diagram, text, rectangle

Description automatically generated

איור 9: הצגת צפיפות ווקטורי המאפיינים של רשת inception עבור מגוון ארכיטקטורות לאחר שעברו הורדת מימדים באלגוריתם PCA. הגרפים המוצגים משמאל לימין, גרף התפלגויות הווקטורים עבור מקבץ תמונת מקור, גרף התפלגויות הווקטורים עבור מקבץ של ממקרה קריסת המצב וגרף מהתפלגויות הווקטורים עבור מקבץ מתוצרי ארכיטקטורת SrPix2Pix שאיתה אימנתי ואתאר בהמשך. ציר ה X וציר ה Y הם ערכי הווקטורים אשר חושבו באלגוריתם PCA בעלי השונות המקסימילת והשניה בגדולה בהתאמה.

## WGAN

מאמר [[6 בא לענות על בעייות הנובעות מהצורך לאזן בין אימוני הגנרטור והדיסקרימינטור, אשר הוסבר בפרק הקודם. לצורך אימון יעיל יותר של רשת הדיסקרימנטור המאמר מציע להחליף את פונקציית ההפסד של הרשת לפונקציית loss Wasserstein. מטרת פונקציית ההפסד המוצעת היא למצוא את הפונקציה אשר תביא במינימום שינוי את התפלגות של מוצא הרשת הגנרטבית להתפלגות האוביקטים מהעולם האמיתי . מרחק הווסרשטיין מספק מדד לכמות ה'"מסה" שיש להזיז, ועד כמה רחוק צריך להזיז אותה, כדי להמיר התפלגות אחת לתפלגות השנייה. הדיסקרימנטור ב-WGAN מתאמן להעריך את מרחק ווסרשטיין.

בהגדרה המקורית של אריכטקטורת GAN, הדיסקרימנטור הוא מסווג, האימון של הדיסקרימינטור לאופטימליות יכול לגרום לבעיה של העלמות גרדיאנטים, משום שהדיסקרימינטור עשוי להיות טוב מדי. דבר זה הופך את הלמידה של הגנרטור לקשה מכיון שלא יהיה משוב מהגרדינאטים. השינוי בפונקציית ההפסד למרחק ווסרשטיין נחוץ מכיוון שהוא מקל על הצורך באיזון האימון בין הדיסקרימנטור לגנרטור. בפונקציית הפסד החדשה, ככל שהערכה של הדיסקרימינטור של מרחק הווסרשטיין טובה יותר, היא יכולה להדריך את הגנרטור בלמידה לייצר דוגמאות שדומות יותר לנתונים האמיתיים. במילים אחרות, הגרדיאנט שבו הגנרטור משתמש לעדכן את הפרמטרים שלו הופך להיות מהימן יותר. מכיוון שכך ניתן לאמן את הדיסקרימנטור בצורה עדיפה. מחברי המאמר מציעים לבצע 5 עדכוני דיסקרימנטור עבור כל אימון גנרטור, גם אני במעבר לארכיטקטורה זו בחרתי לבצע את אותו יחס בין עדכוני גנרטור לדיסקרימנטור.

בחישוב המרחק, על מנת למצוא את הפונקציה המינימלית שתביא למעבר בין ההתפלגויות ניתן להשתמש בדואליות Kantorovich-Rubinstein אשר מגדירה כי לחישוב המטריקה, ניתן למצוא פונקציה f אשר תקבל ערך מקסימלי עבור המרחקים

מפונקציה f ישנה דרישה להיות 1-Lipschitz רציפה ומשמע שהנורמה של הגרדיאנט שלה הוא 1 בכל מקום. הסבר מפורט יותר על מרחק Wasserstein ניתן לראות בעבודה המסכמת שלי במקטע 3.2.

מאמר [6] דואג לכך שפונקציה f תהיה 1-Lipschitz על ידי שינוי בעדכון משקולות בכל איטרציה, על ידי ביצוע תחימה של הערכים להיות בטווח מצומצם. פעולה זו גורמת לכך שהמשקולות יהיו מוכלות רק במרחב קטן ובכך ניתן לשמור את f להיות  1-Lipschitz . המאמר מציין ששיטה זו של תחימת המשקולות בעייתית מכיוון שקשה להגדיר מה התחום הנכון כפי שנסביר בהמשך בעבודה זו.

הקוד עבור ארכיטקטורה זו כתוב במחלקה SrWgan. השינוי אשר בוצע ביחס למחלקת SrGan הוא בשינוי פונקציית האימון לדסקרימנטור ולגנרטור אשר ייגרמו לדיסקרימינטור לחשב את מרחק ווסרשטיין. בנוסף, לאחר כל צעד לשמירה תנאי ליפשיץ אני מבצע את התחימה למשקולות עבור כל השכבות, השתמשתי בערכי תחימת משקולות כפי שהוצע במאמר [6]. תחום הערכים הוא [0.01,0.01-[.

בגרפים למטה מוצגים ההבדלים בין תוצרי הארכיטקטורות כאשר ארכיטקטורת SrGan המקורית מוצגת בשחור והארכיטקטורה אשר עובדת עם פונקציית הפסד ווסרשטיין SrWgan, מוצגת בתכלת. ניתן לראות כי בכל המדדים להשוואה הארכיטקטורה המשתמשת בווסרשטיין עדיפה. בנוסף ניתן לראות כי האימון יציב ביחס לארכיטקטורת SrGan המקורית, מכיוון שהמדדים משתנים בצורה פחותה, בייחוד FID ו SSIM אשר כמעט ולא משתנים.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור 10: גרפים המציגים את מדדי איכות תוצרי הרשת כפונקציה של התקדמות האימון עבור ארכיטקטורת SrWgan בתכלת וארכיטקטורת SrGan בשחור. משמאל לימין, PSNR, FID, SSIM.

בטבלה למטה מוצגות מספר תמונת אשר נלקחו מדגימת תוצרי הארכיטקטורה, תחילה מזמן 1.4 מליון צעדים ובהמשך בקפיצות של 100 אלף צעדים. התמונות מוצגות משמאל לימין. ככל שמספר הצעדים גדול כך התמונה ימנית יותר. כל זוג שורות מציגות הדגמה באימון עבור תמונה מסויימת, בשורה העליונה ניתן לראות תוצרי רשת הגנרטור בארכיטקטורת SrGan, בשורה התחתונה ניתן לראות את תוצרי רשת הגנרטור עבור ארכיטקטורת SrWGAN . ניתן לראות כי אכן המדדים לאיכות תמונה מייצגים את השינוי בתוצרי הארכיטקטורת. ככל שמתקדמים באימון SrGan בצורותו הרגילה, התוצרים מתרחקים מתמונות המקור, ונוצרים פגמים בתמונה. לעומת זאת בתוצרים של האריכטוקטורה SrWGAN, נראה כי חל שיפור ביציבות האימון, ונוצרים פחות פגמים לאורך זמן.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1.8 | 1.7 | 1.6 | 1.5 | 1.4 | Arch/steps |
|  |  |  |  |  | SrGan |
|  |  |  |  |  | SrWGAN |
| A picture containing human face, person, lip, clothing  Description automatically generated | A picture containing human face, person, lip, eyebrow  Description automatically generated | A person with long brown hair  Description automatically generated with low confidence | A person with brown hair  Description automatically generated with low confidence | A close up of a person's face  Description automatically generated | SrGan |
| A person with brown hair  Description automatically generated with low confidence | A person with brown hair  Description automatically generated with low confidence | A person with brown hair  Description automatically generated with low confidence | A person with brown hair  Description automatically generated with low confidence | A person with long brown hair  Description automatically generated with low confidence | SrWGAN |
| A close-up of a person  Description automatically generated | A person with blonde hair  Description automatically generated with low confidence | A close-up of a person  Description automatically generated | A person with blonde hair  Description automatically generated with low confidence |  | SrGan |
| A close-up of a person's face  Description automatically generated | A close-up of a person  Description automatically generated | A close-up of a person  Description automatically generated | A close-up of a person's face  Description automatically generated | A close-up of a person  Description automatically generated | SrWGAN |

איור 11: הדגמת תוצרי ארכיטקטורות SrGan וSrWgan לאורך התקדמות האימון משמאל לימין.

## WGAN-GP

כותבי מאמר [7] Improved Training of Wasserstein GANs הקלו על האימון של WGAN בכך שהורידו את הצורך בקביעת ערך לתחימה של המשקולות בכל שלב באימון כפי שנעשה ב WGAN. השיפור עוזר מפני שגודל התחימה של המשקולות הוא דבר קשה להגדרה, כפי שהסבירו במאמר [6]. תחימה קטנה מידי מקשה על אימון מהיר ותחימה גדולה מידי מקשה על שמירה של תנאי ליפשיץ הנדרש באימון. השוני בWGAN-GP הוא בהוספת תנאי רגולציה לפונקציית ההפסד, אשר עוזר לשמור על תנאי ליפשיץ. בתנאי הרגולציה החדש מודדים את הגרדיאנט של הדיסקרימנטור ביחס לתמונה הנוצרת מאינטרפולציה של שתי תמונות, האחת תמונה הנדגמת מהעולם האמיתי והאחרת תמונה הנוצרת מהרשת הגנרטיבית. ככל שהגרדיאנט גדול יותר, ככה משקל התוספת של תנאי הרגולציה לפונקציית ההפסד יהיה גדול יותר.

יצרתי מחלקה חדשה אשר מממשת את אימון WGAN GP המחלקה מצויה בקובץ sr\_wgan\_gp.py

מימוש תנאי הרגולציה מבוצע בפונקציה gradient\_penatly . המימוש מבצע ראשית יצירת מקבץ תמונות של אינטרפולציה בין תמונות מקור לתמונות אשר נוצרה על ידי הגנרטור, לאחר מכן מחשבים את נורמת הגרדיאנט של הדיסקרימנטור ביחס לבאטץ הזה ומחשבים את העונש לפי המרחק הריבועי מהערך 1, שזהו הערך הגרדיאנט המקסימלי הרצוי עבור תנאי 1 ליפשיץ.

באיור 12, אני מציג את ערכי המדדים לאורך התקדמות האימון עבור ארכיטקטורות SrWgan ו SrWganGp עם כמה ערכי משקל שונים עבור תנאי הרגולציה בפונקציית ההפסד. מבדיקת מטריקות לאיכות תוצרי הרשת נראה כי ארכיטקטורות SrWgan עם פונקציית הפסד ווסרשטיין ותחימת משקלות עבדה בצורה יותר טובה לאורך זמן. מאמר [11] ממשקל את תנאי 10 פעמים ביחס לתנאי המקורי של פונקציית ההפסד במרחק ווסרשיין. בנוסף למשקל תנאי הרגולציה המוצע במאמר, ניסיתי כמה פרמטרים לערכי המשקל לתנאי הרגולציה הנוסף, בכל המקרים, נראה כי אימון SrWgan עם תחימת משקולות היה יותר יציב. הגרף השחור הוא אימון SrWgan עם תחימת משקולות, הגרף התכלת הוא אימון SrWganGp עם משקל 10 אשר זהו הערך הנבחר גם במאמר [11]. הצבעים עבור כל אימון מופיעים באיור 12 , הערכים הנוספים למשקל תנאי הרגולציה הם 2 ו 0.5 אשר מוצגים בגרפים הוורודים והכתומים בהתאמה.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור12 : גרפים המציגים את מדדי איכות תוצרי הרשת כפונקציה של התקדמות האימון עבור ארכיטקטורת SrWgan ו ארכיטקטורת SrWganGp עם כמה משקלים שונים עבור תנאי הרגולציה. משמאל לימין, PSNR, FID, SSIM.

מאמר [11] ממליץ לבטל את שכבת ה batch normalization לדסקרימנטור ובמקום זאת להשתמש ב layer normalization. כאשר מיישמים batch normalization , פלט מסוים הופך להיות תלוי לא רק בקלט התואם אליו אלא גם בכל הקלטים האחרים בקבוצה, מכיוון שהממוצע והשונות מחושבים על גבי כל הקבוצה. לכן, נכון להגיד שנורמליזציה בעזרת batch normalization משנה את הבעיה ממיפוי של קלט אחד לפלט אחד למיפוי מקבוצה שלמה של קלטים לקבוצה שלמה של פלטים. בתנאי רגולציה על הגרדיאנט רצינו להטיל קנס על הנורמה של הגרדיאנט של הדיסקרימנטור ביחס לקלט יחיד, ולא לכל הקבוצה. כדי לפתור את הבעיה הזו, מחברי המאמר הציעו להשמיט את שכבת ה batch normalization ולהחליפו ב layer normalization.

בשלב זה ביצעתי בדיקה כמה שינוי זה משפיע על התוצרים עבור SrWganGp ואכן ישנה השפעה גדולה, התוצרים השתפרו ונהיו דומים ל SrWgan. בדקתי בנוסף האם החלפת שכבת הנורמליציה בגנרטור עוזרת, במקרה זה התוצרים יצאו פחות טובים. להלן הגרפים המציגים את מדדי איכות התמונה הנבדקים לאורך התקדמות האימון. ניתן לראות כי הגרפים אשר משנים את תנאי הרגולציה עבור הדיסקרימנטור)תכלת סגול וכתום) דומים לגרף wgan ואף משפרים אותו באופן מזערי.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור 13: גרפים המציגים את מדדי איכות תוצרי הרשת כפונקציה של התקדמות האימון עבור ארכיטקטורת SrWgan (שחור)ו אריכטקטורת SrWganGp בשינוי שכבות הנורמליזציה עבור הדיסקרימנטור עם כמה משקלים שונים עבור תנאי הרגולציה (כתום, ורוד ,ותכלת) , SrWganGp עבור שינוי שכבת הנורמלציה גם בגנרטור עם כמה משקלים שונים (סגול וירוק) . משמאל לימין, PSNR, FID, SSIM.

בטבלה למטה ניתן לראות את ערכי המדדים למדידת איכות תמונה וזמני הריצה עבור האימון לאחר 3 מיליון צעדים. ניתן לראות SrWgan עבד בצורה מהירה יותר, זאת מכיוון שבחישוב GP אנו קוראים לרשת נוספת לקבלת גרדיאנטים בתמונה, דבר המאט את האימון בצורה משמעותית. ניתן לראות שערכי SrWganGp10 היו הכי טובים ולכן התוצרים מצופים להיות הכי קרובים לתמונות מהעולם האמיתי.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TIME | FID | PSNR | SSIM | ארכיטקטורה |
|  | A picture containing text, screenshot, font  Description automatically generated |  | A picture containing text, screenshot, font  Description automatically generated |  |

## Pix2Pix

ארכיטקטורת Pix2Pix אשר מתוארת במאמר [8], מאפשרת לייצר מיפוי בין תמונות לפי שינוי כללי כרצוננו. מחברי המאמר מתארים מספר שימושים לארכיטקטורה זו לצורך שינויי תמונה אשר עובדים בצורה טובה ללא צורך בהתאמת הארכיטקטורה למשימה הספציפית. להסבר מפורט עבור ארכיטקטורה זו ניתן לקרוא את סעיף 4.3 בעבודה המסכמת. העקרונות העיקריים שארכיטקטורה זו מציגה הם בדסקרימנטור אשר לו הם קוראים PatchGan, שימוש בתנאי רגולציה L1 על שחזור התמונה בפונקציית ההפסד, שימוש בגנרטור מצורת UNET, ושימוש בארכיטקטורת CGAN.

דסקרימנטור Patch Gan מסווג עבור כל מקטע בתמונה האם הגיע מהעולם האמיתי או מהרשת הגנרטיבית. צורה זו של הדיסקרימנטור אשר מחזיר מספר ערכים עבור אזורים שונים מאפשרת לרשת ה GAN ללמוד הרבה יותר מידע מאשר דיסקרימנטור רגיל שמחזיר תשובה כללית אחת לגבי התמונה. בצורה זו ניתנת לרשת היכולת לבחון אזורים ספציפיים כמה הם אמיתיים או לא.

בארכיטקטורת CGAN כפי שמתוארת במאמר [9] מכניסים ווקטור מידע המכיל מידע על ייצור האובייקט הרצוי, ווקטור המידע מוכנס לרשת הדיסקרימינטיבית והגנרטיבית. ווקטור מידע זה מאפשר לרשת הדיסקרימנטיבית לפסול או לאשר תוצרי רשת גנרטור לפי המחלקה של הווקטור. בכך אנו יוצרים התניה של איזה מחלקה אנו רוצים לייצר. בארכיטקטורהPix2Pix משתמשים בעקרון זה, לרשת הגנרטיבית והדסקרימנטיבית מכניסים תמונה כווקטור תנאי, ולפי ווקטור תנאי זה מייצרים את התמונה הרצויה ברשת הגנרטור.

השינויים אשר ביצעתי ליצירת המחלקה SrPix2Pix ביחס למחלקה SrGan הם:

השתמשתי בדסקרימנטור PatchGan אשר על תמונה בגדול 96x96 מחלק את התמונה ל 10x10 מקטעים שעליהם הוא בוחן האם הם הגיעו מהעולם האמיתי או מרשת הגנרטור. אני מציג את ארכיטקטורת הדיסקרימנטור באיור 14.

הוספתי את תנאי L1 לפונקציית ההפסד בין התמונה המוקורית לתמונה המיוצרת. כפי שאתאר בהמשך, בדקתי כמה סדרי גודל של משקל תנאי רגולציה זה לפונקציית ההפסד.

כפי שנעשה במאמר [[10, הכנסתי את התמונה ברזולוציה הגבוהה כווקטור התנאי הנוסף. כאשר בחנתי את השפעת שינוי זה בנפרד על ארכיטקטורה SrGan, התוצאות היו פחות טובות מבחינת המדדים אשר אני מציג לאורך העבודה. בנוסף בדקתי את הארכיטקטורה SrPix2Pix ללא ווקטור התנאי לדסקרימנטור, וגיליתי שהתוצאות זהות לאימון עם ווקטור התנאי לאורך כל האימון.התוצאות מוצגות בגרפים באיור 15 ביחס לגרף SrWgan. הגרפים אשר מציגים את הארכיטקטרות עם ווקטור התנאי נקראים בתחיליות pix2pix, הגרפים אשר מציגים ארכיטקטורות ללא ווקטור התנאי נקראים עם תחילית patch. סוף שם הארכיטקטורה מייצג את גודל משקל תנאי הרגולציה L1, הגדלים שבחנתי עבור משקל זה הם 0.1 10 1.

A diagram of a data flow

Description automatically generated

איור 14: מבנה PatchGan בו השתמשתי בפרויקט. שתי כניסות בגדול 96X96 ויציאה אחת בגדול 10X10

באיור 15, אני מציג גרפים המציגים את מדדי איכות תוצרי הרשת כפונקציה של התקדמות האימון עבור ארכיטקטורת SrPix2Pix וארכיטקטורת SrWgan עם כמה משקלים שונים עבור תנאי הרגולציה L1 עבור ארכיטקטורת SrPix2Pix. ניתן לראות כי מדד FID היה נמוך ודומה לערכי srWgan כאשר השתמשי ב L1 ביחס של 0.1 לפונקציית ההפסד עבור הגנרטור בארכיטקטורות SrGan. בנוסף ערכי ה PSNR ו SSIM היו דומים לכל ריצות ארכיטקטורת SrPix2Pix וגבוהים ביחס לארכיטקטורת SrWgan. לכן ניתן לומר כי הארכיטקטורה עבדה בצורה הטובה ביותר כאשר תנאי הרגולציה L1 בגנרטור הוסף במשקל 0.1 ביחס לפונקציית ההפסד של הגנרטור ב SrGan, בנוסף הארכיטקטורה שיפרה את התוצאות הטובות ביותר שקיבלנו עד שלב זה עבור ארכיטקטורתSrWgan .

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור 15: גרפים המציגים את מדדי איכות תוצרי הרשת כפונקציה של התקדמות האימון עבור ארכיטקטורת SrPix2Pix וארכיטקטורת SrWgan עם כמה משקלים שונים עבור תנאי הרגולציה. משמאל לימין, PSNR, FID, SSIM.

בטבלה למטה ניתן לראות את תוצרי הרשתות הגנרטיביות עבור 3 מיליון צעדים מצד שמאל ארכיטקטורת SrWgan בעמודה האמצעית SrPix2Pix כאשר החלק הסופי בשם הארכיטקטורה מציג את משקל תנאי הרגולציה L1 בגנרטור, בצד ימין מוצגת תמונת המקור.

|  |  |  |
| --- | --- | --- |
| SrWgan | Pix2Pix\_01 | hr |
| A close-up of a person's face  Description automatically generated | A person with brown hair  Description automatically generated with low confidence | A person with dark hair  Description automatically generated with low confidence |
| A person with long hair and beard  Description automatically generated with medium confidence | A person with long hair and beard  Description automatically generated with low confidence | A person with long hair and beard looking to the side  Description automatically generated with low confidence |
| A person with a goatee  Description automatically generated with low confidence | A person with a beard  Description automatically generated with low confidence | A picture containing human face, person, person, clothing  Description automatically generated |
| A close-up of a person's face  Description automatically generated | A close-up of a person  Description automatically generated | A person with blonde hair  Description automatically generated with low confidence |

איור 16: הדגמת תוצרי רשת הגנרטור עבור ארכיטקטורות SrWgan ו SrPix2Pix (עמודה שמאלית ומרכזית) למול תמונות מהעולם האמיתי (עמודה ימנית)

נסיון נוסף שביצעתי הוא לבדוק האם נוצרים מקרים של קריסות מצב כאשר משתמשים בארכיטקטורת SrPix2Pix במשקלים לא מאותחלים לאחר transfer learning כפי שבוצע באימוני הארכיטקטורות עד שלב זה, והאם עדיף להתחיל את האימון ממשקלים מאופסים בהתחשב שתנאי הרגולציה L1 על תוצרי הגנרטור ימנע את קריסות המצב.

בטבלה למטה ניתן לראות את תוצאות המדדים לפי התקדמות האימון על ארכיטקטורת Pix2pix במשקל 0.1 על תנאי הרגולציה L1. התוצאות די דומות לאחר 3 מליון צעדים, עם transfer learning וללא.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור 17: הצגת ערכי המדדים להשוואת ארכיטקטורות למול צעדי האימון עבור ארכיטקטורות SrPix2Pix . בתכלת אימון ללא transfer learning. בסגול ושחור אימון עם transfer learning. ניתן לראות שאין הבדל משמעותי בתוצאה הסופית בין הריצות .

## SinGan

מאמר [12] מציג אימון ארכיטקטורה המשתמשת באימון רשתות יריבות אשר מתאמן על תמונה יחידה. המאמר מציג שיטה לייצור כמה שינויים בתמונה היחידה המתקבלת. המאמר מוצג בהרחבה בעבודה המסכמת בסעיף 4.6.

הארכיטקטורה משתמשת באימן progressive growing , אשר עוזר לארכיטקטורה ללמוד מאפייני טקסטורות בכמה רמות. בכל רמה נעשה אימון WGAN GP עם דסקרימנטור PatchGan. לאחר אימון רמה מסוימת הדיסקרימנטור נשאר ללא שימוש והגנרטור משמש לאימון הרמה הבאה כאשר משקולותיו נשארות מוקפאות. ישנן מספר שיטות לאימון progressive growing, העקרון הבסיסי באימון שכזה הוא שמאמנים את המודל השלם בחלקים הולכים וגדלים. הסבר על progressive growing ויתרונותיו ניתן לראות בעבודה המסכמת בסעיף 3.4. במאמר [12] מודל הדיסקרימנטור נשאר בגודל קבוע בכל שלב באימון. לכן גודל המקטעים שהדיסקרימנטור בוחן עליהם נשאר קבוע עבור כל גודל תמונה המיוצרת ברמות השונות, הדבר גורם לכך שכל רמה מקטעי הדיסקרימנטור רואים חלקים הולכים וקטנים מהתמונה. דבר זה גורם לאימון להתמקד בכל שלב ברמת דיוק פרטים שונה. ככל שגודל הפאטץ בדסקרימנטור באופן יחסי יותר קטן בתמונה הנוצרת, הוא מתמקד באובייקטים יותר קטנים ובטקסטורות יותר חדות. ככל שהפאטץ באופן יחסי גדול הוא מתמקד באוביקטים יותר גדולים וקווי מתאר כללים.

לצורך הדגמת ארכיטקטורה זו השתמשתי בpatchGan בו השתמשתי לארכיטקטורת pix2pix, את שלב הprogressive growing ביצעתי פעמיים, לקחתי את אותה מחלקה SrWgnPix2PixHd אשר גרמתי לה לייצר הגדלה של פי 2, עשיתי זאת על ידי הוצאת שכבתupsample אחת ממודל ה resnet איתו עבדתי בפרויקט. ביצעתי 2 הגדולת תמונה פי 2 כלומר אימנתי 2 מודלי גנרטור בזה אחר זה, כאשר כל אימון גנרטור מתחיל מדסקרימנטור מאותחל במשקולות אקראיים, הדיסקרימנטור בכל שלב לא משומש לאימון בשלב הבא.

תוצאות הארכיטקטורה היו פחות טובות מארכיטקטורת pix2pix בכל המדדים כפי שניתן לראות בגרפים, תוצאות הארכיטקטורה הם בצבעים וורוד לארכיטקטורה אשר מקפיאה את המשוקולות בכל עליית רמה ובתכלת לארכיטקטורה שלא מקפיאה את המשוקולת בכל שלבי האימון.

|  |  |  |
| --- | --- | --- |
|  | | |
|  |  |  |

איור 18: הצגת תוצאות SinGan (גרפיים צבונעיים) למול pix2pix (בשחור), ניתן לראת כי אין שיפור בביצועים בארכיטקטורה זו למול pix2pix

## מציאת התמונות עם השיפור הגדול ביותר בין הארכיטקטורות.

לצורך מציאת הבדלים משמעותיים ביותר בין הארכיטקטורות, יצרתי פונקציה שמבצעת חיפוש להפרש השינוי המשמעותי ביותר, מבין כל תמונות המבחן (תמונות אשר לא הוצגו למודל בשלב האימון). הפונקציה מקבלת כפרמטר רשימה של מודלים, כמות תמונות מקסימלית להצגה, ומדד חיפוש ( המדדים הנתמכים הם FID וSSIM), הפונקציה מחפשת את התמונה עליה עבור המדד הנבחר גדול ביותר בין המודלים השונים. הפונקציה נקראת get\_n\_images\_with\_most\_diffrancess והיא כתובה במחלקה שאחראית על הצגת המידע בפרויקט display\_handler. להלן תוצאות ריצת הפונקציה על 2 המדדים, כאשר מודל Pix2Pix היא המודל עבורו רצינו להציג את השיפור. באיור 19 ניתן לראות כי אכן בתמונות המוצגרות ישנו שיפור ניכר בין ארכיטקטורת Pix2Pix לשאר הארכיטקטורת.

|  |  |  |  |
| --- | --- | --- | --- |
| SrGan | SrWGan | SrPix2Pix | HR |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

איור 19: הצגת התמונות עליהן לארכיטקטורות SrPix2Pix היתה הצלחה הגדולה ביותר יחסית לארכיטקטורות האחרות במדדים SSSIM ו FID

## הצגת התפלגות ווקטור המפאיינים ברשת inception

לצורך הדגמת התפלגות המאפיינים הנוצרים בארכיטקטורות השונות, ביצעתי תהליך של הוצאת ווקטור מאפיינים. ווקטור המאפיינים נלקח מתוצאות מעבר תמונה בשכבה פנימית של רשת Inception מאומנת. הצגתי על גבי תמונה דו מימדתית את הווקטורים מהתמונות שקיבלתי בכל ארכיטקטורה.

מכיון שאנו יכולים להציג גרפים בעבודה זו רק בדו מימד, ביצעתי התמרת PCA והצגתי עבור כל מודל שאומן גרפים אשר מכילים את 2 הווקטורים המרכזיים עבור התמרה זו. התמרת PCA מייצרת קומבינציה לינארית של הווקטרים המקוריים כך שהם מוטלים למרחב ווקטורים אורתוגונליים באופן בו הווקטור המרכזי הוא בעל השונות הגדולה ביותר עבור המאפיינים המקוריים, והווקטור השני הוא בעל השונות השניה בגדולה עבור ווקטור המאפיינים.

ניתן לראות את הפילוג של תמונות אמיתיות, ופילוג של תמונות הנוצרות בארכיטקטורות השונות. כפי שתיארנו בחלק של קריסת המצב בפרק 8 במסמך זה, אכן ניתן לראות שבקריסת המצב כל התמונות הנוצרות היו מאותם מאפיינים דומים מכיוון שהערכים מתרכזים רק באזור מצומצם מאוד בגרף.

|  |  |  |  |
| --- | --- | --- | --- |
| Pix2Pix | SrGan | SrGan No Transfer | HR |
|  |  |  |  |

איור 20: הצגת התמונות עליהן לארכיטקטורות SrPix2Pix היתה הצלחה הגדולה ביותר יחסית לארכיטקטורות האחרות במדדים SSSIM ו FID

## סיכום

בפרויקט זה מימשנו ארכיטקטורות GAN המבצעות סופר רזולוציה.

הסברנו עקרונות אימון לארכיטקטורת GAN, הסברנו את האלגוריתם הבסיסי והסיבות לקושי באימון.

מטרת הפרויקט היתה לראות איך שינויים אשר הוצעו במאמרים לארכיטקטורות GAN יכולים להשפיע על תוצאות מאמר [2]. ראינו הרבה שינויים אשר משפרים את הארכיטקטורה המקורית עליה עבדנו כמודל בסיסי SrGan .

במאמר הראנו שימוש במדדים שונים אשר התאימו לתוצאות השונות שקיבלנו.

ניתן לומר כי לדסקרימנטר PatchGan ופונקציית הפסדWasserstain היתה השפעה ניכרת על התוצאות ארכיטקטורת SrGan.

בפרויקט זה היה מספר נושאים שלא נדונו וניתן לחשוב עליהם כהמשך לפרוירקט.

לרוב שמייצרים מאפיינים חדשים, רוצים ליצור מגוון שלהם, בהרבה ממאמרי GAN מתיחסים למידת הגיוון במאפיינים, ניתן לבחון האם אצלנו נוצרים מאפיינים מגוונים חדשים ואיך ניתן להשפיע על מידת הגיוון.

בפרויקט דנו בהגדלת רזולוציה עד פי 4 כפי שבוצע במאמר [2]. ניתן לבחון האם הגדלת רזולוציה בפקטור יותר גדול מצריכה פתרונות אחרים, או שהארכיטקטורות שהצגנו מספקות.

ביצוע שיפור ברזולוציית התמונה הוא אלגוריתם שימושי מאוד, שיפור האלגוריתמים בו יכול להיות בכמה מובנים כגון: מהירות הפעלה, יציבות האימון, דיוק התוצאות וחדות התמונה. כאשר ביצענו השוואות בין ארכיטקטורות בעבודה זו, אלו המאפיינים אשר דנו בהם.

## רשימת מקורות

[1[ Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.‏

[2] LEDIG, Christian, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. p. 4681-4690.

[3] JOHNSON, Justin; ALAHI, Alexandre; FEI-FEI, Li. Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. Springer, Cham, 2016. p. 694-711.

[4] HE, Kaiming, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778.‏

[5] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612

[6] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." International conference on machine learning. PMLR, 2017.‏

[7] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." Advances in neural information processing systems 30 (2017).‏

[8] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.‏

[9] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." arXiv preprint arXiv:1411.1784 (2014).‏

[10] Qiao, Jiaojiao, et al. "Image super‐resolution using conditional generative adversarial network." IET Image Processing 13.14 (2019): 2673-2679.‏

[11] Shi, Wenzhe, et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.‏

[12] SHAHAM, Tamar Rott; DEKEL, Tali; MICHAELI, Tomer. Singan: Learning a generative model from a single natural image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019. p. 4570-4580.‏