

האוניברסיטה הפתוחה
המחלקה למתמטיקה ולמדעי המחשב

אלגוריתמים לשיפור איכות תמונה המבוססים על רשתות GAN

עבודה מסכמת זו הוגשה כחלק מהדרישות לקבלת תואר
"מוסמך למדעים" M.Sc. במדעי המחשב
באוניברסיטה הפתוחה
המחלקה למתמטיקה ולמדעי המחשב

על-ידי
ליאור בומורצל

העבודה הוכנה בהדרכתה של ד"ר מירי אביגל
פברואר 2023

תקציר

אלגוריתמים המשתמשים ברשתות יריבות נפוצים מאוד בקרב משימות של שינוי ערכי תמונה. הסיבה לכך היא שאלגוריתמים אלו מצליחים לייצר תמונות הנראות אמיתיות וחדות. האלגוריתמים לומדים את אופי הטקסטורות מהעולם האמיתי בשלב האימון והם מצליחים לייצר שחזור של טקסטורות אלו בהשלמת המידע ככל שנדרש בשלב ההפעלה.

בעבודה אסקור מגוון אלגוריתמים המשתמשים באימון רשתות יריבות אשר מראים שינויים בתמונה, אתמקד באלגוריתמים המשפרים את מאפייני התמונה במספר מובנים מגוונים כגון הגדלת רזולוציה, העלמת טשטוש שנובע מתזוזה, העלמת ערפל, יצירת סרטון מונפש מתמונה, עריכת תמונה, והחזרת מקטעים שנעלמו מהתמונה.

אימון רשתות יריבות הוא משימה קשה, בעבודה אסביר את הסיבות לכך ולאחר מכן אסקור מגוון אלגוריתמים המציגים שינויים לארכיטקטורה ומשפרים את יכולות הרשתות היריבות. אעמוד על הצורך בשינויים שמגוון האלגוריתמים מציגים באלגוריתמים שאסקור. השינויים שאציג יהיו בצורת האימון, בפונקציית ההפסד, ובמבנה הארכיטקטורה.

בעבודה אציג מגוון ארכיטקטורות אשר לכל אחת מהן מתאים בסיס נתונים מסוג אחר. חלק מהארכיטקטורות צריכות זוגות של תמונות בעלות מיפוי ממקור לתוצר הרצוי. חלק מהארכיטקטורות יכולות לעבוד על זוג מקבצי תמונות כאשר את המיפוי הארכיטקטורה תלמד באופן עצמאי, ארכיטקטורה נוספת שאציג תצליח ללמוד לייצר מגוון שינויים בתמונה מתמונה יחידה ללא בסיס נתונים נוסף.

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. אין הגדרה מסויימת איך התמונות הסופיות בתוצרי הרשת אמורים להראות, לכן השוואה בין ארכיטקטורות לרשתות יריבות זו משימה קשה. במסגרת העבודה אסקור מגוון אלגוריתמים ומדדים למדידת טיב איכות התוצרים של הארכיטקטורות השונות, המדדים השונים נדרשים לצורך השוואה בין סוגי תוצרי הארכיטקטורות.

תוכן עניינים

4	1. מבוא.....
6	2. ארכיטקטורת GAN.....
9	2.1 פונקציית ההפסד של הרשת:.....
11	2.1.1 הנקודה האופטימלית של פונקציית ההפסד.....
14	2.2 אימון הרשתות:.....
17	2.3 בעיות נפוצות הנוצרות במהלך אימון רשת GAN.....
23	3. סקירת ארכיטקטורות ושיטות אימון אשר משפרות את אימון הרשת.....
23	3.1 unrolled generative adversarial networks.....
25	3.2 WGAN - Wasserstein GAN.....
31	3.3 WGAN-GP- gradient penalty.....
32	3.4 progressive growing of gans.....
35	4 סקירת ארכיטקטורות ושיטות מובילות לשינוי ערכי תמונה.....
35	4.1 שימוש בפונקציות הפסד לשמירה על מבנה התמונה.....
38	4.1.1 דוגמה לשימוש בפונקציית הפסד לשמירה על תוכן התמונה.....
42	4.2 CGAN - Conditional Generative Adversarial Nets.....
43	4.2.1 דוגמא שימוש בארכיטקטורת CGAN לצורך שיפור איכות תמונה.....
44	4.3 Pix2Pix שינוי ערכי תמונה עבור מיפוי מוצג.....
48	4.3.1 דוגמא לשימוש patchGan.....
50	4.4 CycleGan שינוי ערכי תמונה עבור מיפוי אשר נלמד על ידי הארכיטקטורה.....
52	4.4.1 דוגמה לשימוש בארכיטקטורת CycleGan וב perceptual loss.....
53	4.5 ייצור שינויים נשלטים בתמונה על ידי שימוש בשכבה הלטנטית.....
56	4.6 SinGan – אימון גנרטיבי על תמונה יחידה.....
60	5 הערכת תוצאות רשתות ה GAN.....
61	5.1 PSNR.....
62	5.2 SSIM.....
64	5.3 שימוש ברשתות נוספת למדידת גיוון ודיוק התוצרים.....
64	5.3.1 מדידת טיב זיהוי בעזרת רשתות לזיהוי עצמים.....
64	5.3.2 FCN score.....
66	5.3.3 inception score.....
67	5.3.4 Fréchet Inception Distance (FID).....
69	6. סיכום.....
71	7. ביבליוגרפיה.....

1. מבוא

שיפור איכות תמונה נפוץ היום במגוון רחב של אפליקציות. שיפור איכות תמונה מתבטא בהפיכתה לברורה יותר, והדגשה של האובייקטים עליהם רצו להתמקד בתמונה. לצורך כך ניתן לבצע שינויים במגוון מאפיינים בתמונה, לדוגמה ניתן לבצע הבהרת תמונה, הורדת טשטוש ורעש, שיפור רזולוציה, הורדת השפעה של תנאי מזג אוויר קשים, צביעת תמונה והשלמה של חלקים חסרים בתמונה.

בכל קטגוריה לשיפור תמונה ישנן כמה ארכיטקטורות אשר פותחו ומפותחות בימים אלו. בין האלגוריתמים החדשניים בולט השימוש ברשתות נוירונים ובפרט בארכיטקטורות Generative Adversarial Networks [1] בראשי תיבות GAN. מאז מאמר [1] בו ארכיטקטורת רשתות GAN הודגמו לראשונה, התפרסמו עוד מגוון רחב של ארכיטקטורות אשר כולן העקרון הבסיסי של שימוש ברשתות יריבות המתחרות ביניהן. מטרת האחת (הרשת הגנרטיבית) הוא ליצור תמונה, ומטרת השנייה (הרשת הדיסקרימינטיבית) לסווג האם התמונה באה מהעולם האמיתי או יוצרה על ידי הרשת הגנרטיבית.

לעומת רשתות GAN, ישנן רשתות אחרות המנסות לייצר תמונות בשיטת מזעור שגיאת השחזור MSE. רשתות אלו מייצרות את התוצאה אשר עבורה הפיקסלים הנבחרים יקבלו את המינימום של ממוצע השגיאה הריבועית מבין כל האפשרויות עבור אותם פיקסלים. רשתות המשתמשות בשיטה זו, לרוב מייצרות תמונות מטושטשות. יצור התמונות המטושטשות קורה מכיוון שלפעמים עבור אזור מסוים של פיקסלים ישנן כמה אפשרויות, לכן יכול להיבחר ערך להשלמת הפיקסלים אשר לא בהכרח קיים בתחום תמונות המקור. הסיבה לבחירת ערך זה יכולה להיות מכיוון שזהו הערך אשר ממזער את תוצאת MSE. לרוב הערך אשר יבחר עבור MSE הוא הערך הממוצע עבור כל האפשרויות.

כפי שנראה בעבודה זו, שימוש ברשתות GAN לצורך שיפור איכות תמונה מניב תמונות חדות וברורות. הסיבה ליצירת תמונות חדות בארכיטקטורת ה GAN היא שבשלב האימון מתמקדים ביצירת תמונה אשר עליה דסקרימנטור לא יוכל להבדיל האם הגיעה מהעולם האמיתי או יוצרה על ידי הרשת הגנרטיבית. בהנחה שבבסיס הנתונים שלנו לתמונות המקור ישנן בעיקר תמונות חדות, תמונות מטושטשות יידחו על ידי הדיסקרימנטור מכיוון שייחשבו למזויפות. עקרון זה דוחף את הרשת הגנרטיבית לייצר טקסטורות ממרחב העולם האמיתי, על מנת שיגרמו לדסקרימנטור לטעות על התמונה בהסתברות גבוהה, כפי שניתן לראות באיור מספר 16 בעבודה זו. נדגיש כי, אלגוריתמים לשיפור איכות תמונה המשתמשים ברשתות GAN נעזרים במידע שנצבר במרחב הלטנטי שהרשת לומדת על מנת להשלים את המידע ליצירת תמונת המוצא, כלומר אנו דורשים מהמודל בשלב האימון ללמוד התפלגות של מאפיינים של אובייקטים אמיתיים על פי בסיס הנתונים שהתקבל ובשלב השימוש (לאחר האימון) אנו משתמשים במידע הנלמד להשלמת מאפיינים לאובייקטים בתמונת המוצא. כך על ידי שימוש באלגוריתמים אלה, התוצר הוא לא ניסיון של שחזור או השלמה מדויקת של הפרטים של המידע (אשר לפעמים לא קיים), התוצר הוא ניסיון של הרשת הגנרטיבית לטייב את התמונה בצורה

כזו שהתמונה הנוצרת תצליח להראות אמיתית עד כדי שתגרום לדסקרימינטור לסווג בהסתברות גבוהה שהתמונה הגיעה מהעולם האמיתי. ככל שיהיו לנו יותר פרטים מתמונת המקור אשר נרצה להעביר לתמונת היעד כך נצטרך להשלים פחות מידע בתמונת היעד על ידי רשת ה-GAN.

משימה גנרטיבית היא משימה קשה יחסית בהשוואה למשימות כגון סיווג, הסיבה לכך היא שאנו מבקשים מהמודל שלנו שידע לייצר אובייקטים מהמחלקה, ובהתפלגות שאנו רוצים לייצר, זאת בניגוד לסיווג אשר שם נדרשת מהמודל רק יכולת הפרדה בין המחלקות.

למרות שרשתות GAN הראו תוצאות מרשימות ביצירת תמונות חדות בשנים האחרונות, בפועל אימון הארכיטקטורות השונות הוא משימה קשה, כפי שנסביר בסעיף 2.3 בעבודה זו. התכנסות המודל, כמות התיגוג וגודל בסיס הנתונים הנדרש תלויים רבות בארכיטקטורה הנבחרת, בצורת האימון, ובבחירת ההיפר פרמטרים השונים. בעבודה נתאר בהרחבה מגוון ארכיטקטורות, ואת הצורך בכל שינוי שארכיטקטורה מציגה. בפרט נתאר את המרכיבים שעוזרים למודלי רשת ה-GAN השונים להתמודד עם המשימה הקשה של אימון לא יציב ולהימנע מהתוצאות הלא רצויות כגון ייצור אובייקטים עם פיצ'רים לא הגיוניים, וייצור מספר אובייקטים מצומצם. בעבודה אציג מספר שינויים באלגוריתמי GAN לצורך התמודדות עם הבעיות המוזכרות, השינויים שאציג יהיו בצורת האימון כפי שנעשה במאמר [23], בפונקציית ההפסד כפי שנעשה במאמרים [3] [11], ובמבני הרשת כפי שאסקור בכמה מאמרים בפרק 4.

בפרק 4 אסקור מגוון סוגי אלגוריתמים לשיפור ערכי תמונה, האלגוריתמים המוצגים בעבודה נבחרו על מנת להראות מגוון ארכיטקטורות שונות של אימון רשתות יריבות אשר דורשות סוגים שונים של בסיסי נתונים. חלק מאלגוריתמים המוצגים כגון [7] צריכים לקבל לאימון בסיס נתונים אשר מכיל זוגות של תמונות הקשורות בפונקציית מיפוי מתמונת המקור לתמונת התוצר הסופי לאימון. חלק מהאלגוריתמים המוצגים כגון [13] צריכים לקבל בבסיס הנתונים לאימון זוג מקבצי תמונות ללא התאמה בין התמונות כלומר נותנים לארכיטקטורה ללמוד את המיפוי בין התמונות מהמקבצים בעצמה. סוג אלגוריתמים נוסף המוצג ב[29] דורש אימון בעזרת תמונה יחידה, כלומר לא צריך להכין בסיס נתונים במיוחד למשימה. כפי שאתאר, להרבה מהאלגוריתמים בפרק 4 יש יכולת לבצע משימה כללית של שינוי ערכי תמונה והבחירה בין האלגוריתמים יכולה להיות על פי זמן לאימון, גודל המודל, ובסיס הנתונים שלרשותנו.

לאחר שינוי התמונה ישנו צורך בהבנה של עד כמה התמונה איכותית. מדידה שכזו הכרחית בהבנה של עד כמה האלגוריתם לשיפור איכות תמונה מוצלח. מדידה של אלגוריתמים המשתמשים בארכיטקטורת GAN היא משימה קשה ופתוחה, הסיבה לכך שאין הגדרה לאיך תוצרי הרשת הגנרטיבית צריכים להראות באופן מוחלט.

לצורך השוואת תוצאות רשתות GAN, רוב המאמרים נעזרים ב mean opinion score - זהו מדד אשר משתמש בדעת קהל הממוצעת לגבי טיב התמונות, שימוש בדעה של אנשים הוא מצרך יקר, לכן מנסים להיעזר בשיטות אוטומטיות למדידת טיב התמונה והמודל. בעבודה זו נתאר את השיטות

האוטומטיות הנפוצות והחדשניות ביותר בפרק 5, בפרט נציג אלגוריתמים ומדדים להשוואה בין תוצרי רשתות GAN, והיתרונות עבור כל אלגוריתם השוואה.

2. ארכיטקטורת GAN

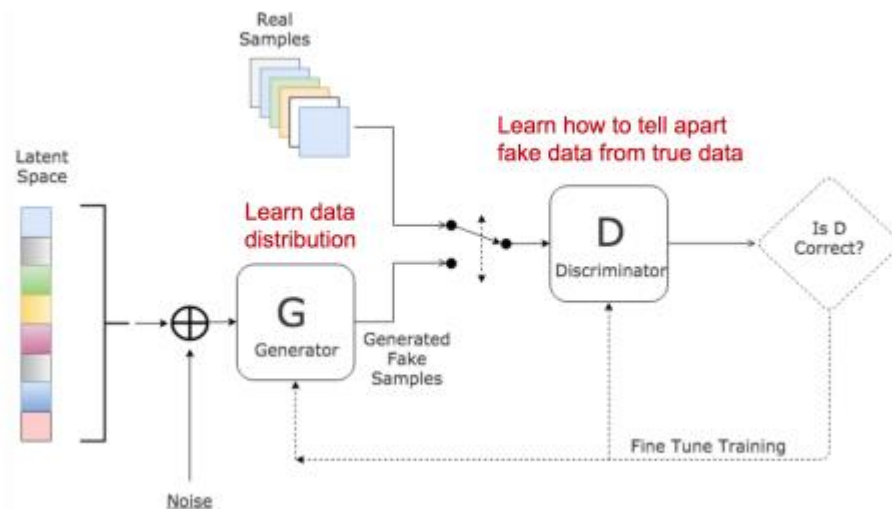
Generative Adversarial Networks זוהי ארכיטקטורה אשר התפרסמה על ידי Ian Goodfellow בשנת 2014, הארכיטקטורה כוללת שתי רשתות יריבות הנקראות דיסקרימיננטור וגנרטור אשר מטרותיהן הפוכות אחת לשנייה.

הארכיטקטורה בנויה בצורה כזאת שהכניסה של הרשת הדיסקרימינטיבית יכולה להגיע משני מקורות, האחד הוא בסיס הנתונים אשר מהווה דוגמא לאובייקטים אמיתיים, והשני הוא האובייקטים הנוצרים מהרשת הגנרטיבית. המוצא של הרשת הדיסקרימינטיבית הוא הסתברות עבור כל אובייקט האם הוא הגיע מבסיס הנתונים או מתוצרי הרשת הגנרטיבית. מטרת הרשת הדיסקרימינטיבית הוא לסווג נכונה את מקור ההגעה של האובייקט בכניסה.

הרשת הגנרטיבית מקבלת בכניסה ווקטור רעש אשר לרוב נדגם בהתפלגות נורמלית או אחידה. על בסיס ווקטור הרעש, הרשת הגנרטיבית יוצרת אובייקטים אשר מטרתם לגרום לרשת הדיסקרימינטיבית לסווג בצורה לא נכונה, ולגרום לה לסווג בהסתברות גבוהה לכך שהאובייקט הגיע מבסיס הנתונים ולא מתוצרי הרשת הגנרטיבית. הסיבה לשימוש בווקטור רעש בכניסה לרשת הגנרטיבית הוא על מנת לייצר דוגמאות חדשות מווקטורי רעש שונים, בהמשך נרחיב ונראה כי במאמרים מתקדמים נמצא שרשת ה GAN לומדת מרחב סמוי (מרחב לטנטי) אשר הבנה שלו תאפשר לבחור ווקטור רעש שבעזרתו נוכל להחליט איזה אובייקטים ייווצרו.

בשלב האימון הארכיטקטורה, הרשת הגנרטיבית מקבלת משוב בעזרת אלגוריתם

Back propagation עבור איזה מהתוצרים מקבלים ערכים גוברים ברשת הדיסקרימינטיבית על פי פונקציית ההפסד. עדכון המשכולות לצורך הקטנת ערך השגיאה נעשה בעזרת אלגוריתם gradient descent



איור 1: מבנה רשת GAN, התמונה למעלה נלקחה מ [2]. בתמונה ניתן לראות את החיבור בין הרשתות הדסקרימנטריות והגנרטיביות. וקטור רעש מוכנס לרשת הגנרטור אשר מייצרת דוגמאות שנכנסות לסירוגין לרשת הדסקרימנטור ביחד עם דוגמאות מהעולם האמיתי. הרשת הדסקרימנטריות מחזירה הסתברות להגעת הקלט מהעולם האמיתי. מטרת הארכיטקטורה היא לאמן מודל גנרטיבי אשר מייצר את התמונות אשר יגרמו למודל הדסקרימנטריות לטעות.

רשת GAN כמודל למידה לא מונחית

בלמידת מכונה, תהליך הלמידה של המערכת נקרא "אימון". בשלב האימון, מציגים למערכת את הדוגמאות והמערכת לומדת היפותזה שמתארת בצורה הטובה ביותר את הדוגמאות שראתה. ניתן לחלק את סוגי האימון ללמידה מונחית ולמידה לא מונחית.

בלמידה מונחית, כל דוגמה שהמערכת לומדת היא זוג, המורכב מאובייקט קלט והפלט הרצוי בעבור אותו הקלט. לאוסף הדוגמאות קוראים בשם "סט אימון". סט האימון מכיל דגימות שמגיעות מהתפלגות משותפת של מרחב האלמנטים ומרחב הפלט הרצוי. מטרת המערכת בלמידה מונחית היא ללמוד היפוטזה למיפוי בין קלטים לפלטים.

בלמידה בלתי מונחית מנסים ללמוד את תכונות לגבי אוסף הדוגמאות המתקבל באימון, כאשר הנתונים לדוגמאות זמינים כפי שהם ללא תוספת תיוג.

רשת GAN הוא מודל למידה לא מונחית מכיוון שאנו לא מנסים לאמן את המודל לייצר לנו תווית של תיוג לאובייקטים או אפילו את האובייקטים עצמם מאובייקטים מתויגים, רשת GAN מייצרת לנו אובייקטים הבנויים מפיצ'רים כלשהם בהתפלגות מסוימת, אשר אנו שואפים שהתפלגות זו תהיה קרובה להתפלגות מאפייני אובייקטים מהעולם האמיתי. אומנם רשת GAN משתמשת במסווג שאומן בשיטת מונחית אשר יסווג לנו האם דוגמאות באו מהעולם האמיתי או יוצרו על ידי הגנרטור אבל התוצאה של רשת GAN שאנו משתמשים בה היא האובייקטים שנוצרו ללא הנחיה ברשת הגנרטיבית.

מודלים דיסקרימינטיביים –

מודלים שמתוכננים לסווג דוגמאות בין מחלקות, על מנת לדעת איך לשייך אובייקטים למחלקות. מודלים אלו צרכים ללמוד את הגבול בין המחלקות. חלק מהמודלים הדיסקרימינטיביים מייצרים מישור מפריד במרחב (או מספר מישורים מפרידים). המישור מפריד את הדוגמאות ששייכות למחלקה אחת מאלו ששייכות לאחרות. במקרה כזה, תהליך הסיווג יכול מציאת אזור במרחב שבו הדוגמה נמצאת והתשובה תינתן לפי המחלקה שמזוהה עם אזור במרחב זה. מודל זה הוא מודל שמטרתו למצוא פונקציית הסתברות אשר בהינתן פיצ'רים מסוימים עבור אובייקט, המודל מחזיר הסתברות לערכי המחלקה אליהם האובייקט שייך כלומר y -class, x -features $p(Y|X)$.

מודלים גנרטיביים –

מטרת מודלים גנרטיביים היא ללמוד ממקבץ דוגמאות הניתנות בהתפלגות מסוימת בשלב האימון, לייצר דוגמאות נוספות מאותה התפלגות.

לרוב, בבניית מודלים גנרטיביים בשיטה הסטטיסטית, מנסים ללמוד את התפלגות האובייקטים מהעולם באופן מפורש. לאחר מכן, ליצירת איבר חדש, ניתן לדגום איבר מההתפלגות אשר נלמדה. מודלים מסוג זה מנסים ללמוד על פי הדוגמאות אותם הם קיבלו, איך מתפלגות דוגמאות מהעולם. לרוב מודלים אלו ישתמשו בשיטה של מציאת הנראות המקסימלית להתאמה בין ההתפלגות הנלמדת והאובייקטים אשר נדגמו מהתפלגות P_r . בדרך זו ייבחרו סט היפוטזות לפונקציות התפלגות אשר נשלטות על ידי פרמטר מסויים ויכולות לתאר את ההתפלגות של האובייקטים מהעולם האמיתי P_r . לאחר מכן, פונקציית ההתפלגות של המודל הגנרטיבי הנלמדת P_g תהיה זו שנבחרה מסט ההיפוטזות ופונקציית ההתפלגות אשר היתה קרובה באופן מקסימלי לסט הערכים הנצפה. על פי מאמר [3] באופן אסימפטוטי בשיטת הנראות המקסימלית אנו מחפשים את הפונקציה שתקטין את ערך $D_{kl}(P_r||P_g)$, ערך Kullback-Leibler (KL) divergence המתאר את המרחק בין ההתפלגויות והוא מוגדר על ידי :

$$D_{KL}(P_r||P_g) = \int_{x=-\infty}^{x=\infty} \left(P_r(x) \log \left(\frac{P_r(x)}{P_g(x)} \right) \right) dx$$

ערכי KL מקבלים ערכים אינסופיים עבור מקומות שבהם ההתפלגות $P_g(x)$ מתאפסת אך לא התפלגות $P_r(x)$. לטיפול בבעיה זו לרוב מוסיפים רעש בערכי P_g , אשר יגרום לכך שלא יהיו מקרים בהם KL מגיע לערכים אינסופיים. הבעיה בהוספת רעש שכזה היא שהוא גורם לייצור התמונה להיות מטושטש.

ארכיטקטורת GAN כמודל גנרטיבי

מטרת האימון של ארכיטקטורת GAN הוא לייצר מודל, הרשת הגנרטיבית. בקבלת ווקטור רעש מהתפלגות P_z המודל ייצר אובייקטים בהתפלגות P_g אשר יהיו קרובים לאובייקטים בעלי מאפיינים

זהים לאובייקטים מבסיס הנתונים הנלמד בהתפלגות P_r . על ידי אימון הרשת הגנרטיבית ושינוי המשקולות, הארכיטקטורה לומדת לשנות את המיפוי ולקרוב את ההתפלגות P_g להתפלגות P_r .

המודל לומד לעשות כך ללא צורך בלמידת ייצוג מפורש של פונקציית התפלגות האובייקטים מהעולם האמיתי P_r .

ייתרון ללמידה שכזאת הוא שלפעמים ייצוג ההתפלגות P_r הוא משימה קשה, והיא לא הדבר אשר מעניין אותנו. לרוב נרצה רק את האפשרות לייצר דוגמאות. במקום לשערך את P_r הארכיטקטורה לומדת פונקציית מיפוי מווקטור הרעש לאובייקטים בהתפלגות P_g .

רשת GAN מכילה 2 רשתות מתחרות: הרשת הדסקרימנטיבית והרשת הגנרטיבית, אשר מטרתן הפוכה. הרשת הדסקרימנטיבית שמטרתה היא לסווג נכון האם תמונה הגיע מהעולם האמיתי או מהרשת הגנרטיבית. הרשת הגנרטיבית מאומנת לייצר אובייקטים אשר ייגרמו לרשת הדסקרימנטיבית להחזיר בהסתברות גבוהה שהגיעו מהעולם האמיתי. ככל שהרשת הדסקרימנטיבית תתקדם באימון ותהיה בעלת יכולת סיווג טובה לגבי האם אובייקט הוא מהעולם אמיתי, כך הרשת הגנרטיבית תצטרך לייצר אובייקטים שיותר דומים למחלקות שהרשת הדסקרימנטיבית לומדת לסווג עליהם נכונה.

ניתן להגדיר את ארכיטקטורת GAN כמודל גנרטיבי מכיוון שהשימוש בתוצרים של הרשת הגנרטיבית הוא הדבר בו משתמשים לאחר האימון. תוצרי רשת ה GAN הם אובייקטים חדשים אשר התפלגותם צריכה להיות כהתפלגות האובייקטים מבסיס הנתונים הנלמד.

2.1 פונקציית ההפסד של הרשת:

לצורך בעיות סיווג נהוג להשתמש בפונקציית הפסד binary cross entropy או BCE בראשי תיבות.

$$LogLoss = -\frac{1}{m} \sum_{i=1}^m [y^i \log(P(x^i)) + (1 - y^i) \log(1 - P(x^i))]$$

פונקציית הפסד זו מודדת את חוסר ההתאמה בין ההתפלגויות של ערכי התוויות Y לתוצאת התפלגות הסיווגים $P(X)$. השימוש באלגוריתם גורם לפונקציית הפסד זו להעניש יותר ככל שהמודל יותר בטוח על תוצאות $P(X)$ אשר אינן נכונות לפי התיוג.

כפי שתיארנו, מטרת הרשת הדסקרימנטיבית לסווג האם דוגמה הגיע מהעולם האמיתי, ומטרת הרשת הגנרטיבית ליצור דוגמאות אשר יגרמו לטעות בסיווג של הרשת הדסקרימנטיבית.

כאשר מחברים את 2 המטרות המנוגדות של הרשת מקבלים משחק MINMAX אשר הערך האופטימלי שלו מגדיר את פונקציית ההפסד:

$$\operatorname{argmin}_G \operatorname{argmax}_D L(G, D) = \frac{1}{m} \sum_{i=1}^m [y^i \log(D(x^i)) + (1 - y^i) \log(1 - D(G(z^i)))]$$

את פונקציית ההפסד ניתן לפתח בנוסף לצורה הבאה:

$$\begin{aligned} \frac{1}{m} \sum_{i=1}^m [y^i \log(D(x^i)) + (1 - y^i) \log(1 - D(G(z^i)))] \\ = E_{x \sim P_r}(\log(D(x))) + E_{z \sim P_z}(\log(1 - D(G(z)))) \\ = E_{x \sim P_r}(\log(D(x))) + E_{x \sim P_G}(\log(1 - D(x))) \end{aligned}$$

y^i מתאר את ערך התיוג של דוגמא האם היא נלקחה מהעולם האמיתי

x^i מתאר פיצ'רים של אובייקט

z^i מתאר ווקטור רעש

G מתאר תוצאת הפעלת הרשת הגנרטיבית

D מתאר תוצאת הפעלת הרשת הדסקרימניבית

$x \sim P_z$ התפלגות וקטור הרעש

$x \sim P_g$ התפלגות האובייקטים המיוצרים מהרשת הגנרטיבית

$x \sim P_r$ התפלגות האובייקטים מדוגמאות המקור

ניתן לתאר את הפונקציה כערך הממוצע של גודל השגיאה של הרשת הדסקרימניבית.

עבור דוגמאות מהעולם האמיתי $y^i = 1$ החלק שלא יתאפס יהיה החלק השמאלי והוא יקבל ערך גבוה ככל שהפרדיקציה של הרשת הדסקרימניבית D על x^i תהיה קרובה ל 1, באותו אופן בכיוון השני כאשר דוגמא יוצרה על ידי הרשת הגנרטיבית $y^i = 0$ והחלקה שלא יתאפס במשוואה הוא החלק הימני אשר יקבל ערך גבוה ככל שהרשת הדסקרימניבית תיתן ערך שהוא קרוב ל 0 על פרדיקציה עבור $G(z^i)$. מטרת הרשת הדסקרימניבית לגרום לפונקציית ההפסד להיות גדולה ככל הניתן הרשת הדסקרימניבית תאומן לתת סיווגים כמה שיותר קרובים לתווית התיוג.

מהצד השני, המטרה של הרשת הגנרטיבית היא למזער את תוצאת פונקציית ההפסד, הרשת הגנרטיבית יכולה להשפיע רק על דוגמאות שהיא מייצרת, לכן היא תרצה למקסם את החלק הימני

במשוואה, ולגרום לכך שהתוצאה של $D(G(z^i))$ תהיה קרובה ל 1 ככל הניתן. הדרך של הרשת הגנרטיבית להגיע לתוצאה של פונקציית הפסד נמוכה היא בכך שהיא תייצר אובייקטים הנראים דומים לאובייקטים מהעולם האמיתי. הרשת הגנרטיבית יוצרת אובייקטים שונים לפי ווקטור רעש המוכנס לה, במאמרים מתקדמים נמצא כי רשת הגנרטור לומדת מרחב סמוי (מרחב לטנטי) אשר הבנה שלו תאפשר לבחור איזה אובייקטים ייווצרו, נרחיב על עקרון זה בהמשך.

2.1.1 הנקודה האופטימלית של פונקציית ההפסד

בחלק זה נראה כי כאשר הדיסקרימינטור הגיע לערך האופטימלי עבורו בפונקציית ההפסד, הערך האופטימלי בפונקציית ההפסד עבור הגנרטור מיוצג על ידי הפונקציה:

$$\operatorname{argmin}_G L(G, D^*) = D_{js}(P_r || P_g) - 2\log 2$$

כאשר D_{js} מייצג מדד Jensen–Shannon divergence המודד מרחק בין התפלגויות הסתברות, המבוסס על Kullback–Leibler divergence.

הגדרת $D_{kl}(P || Q)$: (לא סימטרי ולא מקיים את אי שוויון המשולש ולכן לא מטריקה) זהו מדד אשר מודד את המרחק של P מ Q

$$D_{KL}(P || Q) = \int_{x=-\infty}^{x=\infty} \left(p(x) \log \left(\frac{p(x)}{q(x)} \right) \right) dx$$

הגדרת $D_{js}(P || Q)$: מטריקה המודדת כמה התפלגויות דומות אחת לשניה

$$D_{js}(P || Q) = \frac{1}{2} D_{kl}(P || \frac{P+Q}{2}) + \frac{1}{2} D_{kl}(Q || \frac{P+Q}{2})$$

בהנתן הביטוי שנקבל בפונקציית ההפסד עבור רשת הגנרטור כאשר הדיסקרימינטור מגיע לערך אופטימלי ביותר עבורו בפונקציית ההפסד, $\operatorname{argmin}_G L(G, D^*)$, הערך קטן כאשר ייקטן הערך $D_{js}(P_r || P_g)$ עד לאיפוס, איפוס ערך זה מתקבל כאשר ההתפלגויות P_r, P_g בו זהות. כאשר ההתפלגויות יהיו זהות, פונקציית ההפסד תהיה שווה ל $\operatorname{argmin}_G L(G, D^*) = -2\log$, זהו ערך פונקציית ההפסד לגנרטור אופטימלי כאשר הדיסקרימינטור הגיע לערך האופטימלי שלו בפונקציית ההפסד.

כפי שתיארנו בסעיף 2.1, פונקציית ההפסד של רשת GAN

$$\begin{aligned}
\argmin_G \argmax_D L(G, D) &= \frac{1}{m} \sum_{i=1}^m y^i \log(D(x^i)) + (1 - y^i) \log(1 - D(G(z^i))) \\
&= E_{x \sim P_r}(\log(D(x))) + E_{z \sim P_g}(\log(1 - D(G(z)))) \\
&= E_{x \sim P_r}(\log(D(x))) + E_{x \sim P_g}(\log(1 - D(x))) \\
&= \int_{x=-\infty}^{x=\infty} (p_r(x)(\log(D(x))) + p_g(x)(\log(1 - D(x)))) dx
\end{aligned}$$

נקבע את הדיסקרימינטור להיות בעל הפרמטרים האופטימליים שלו ונחשב את הנקודה האופטימלית של הרשת הגנרטור עבור פונקציית הפסד זו:

$$\argmin_G L(G, D^*) = \int_{x=-\infty}^{x=\infty} (p_r(x)(\log(D^*(x))) + p_g(x)(\log(1 - D^*(x)))) dx$$

מטענת העזר שנוכיח בהמשך נקבל כי הערך האופטימלי עבור הדיסקרימינטור אשר ממקסם את האינטגרל עבור כל ערכי x האפשריים, הוא

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}$$

זהו הערך האופטימלי עבור הדיסקרימינטור ולכן הערך האופטימלי של פונקציית הגנרטור כאשר מתקבל דסקרימינטור אופטימלי הוא:

$$\argmin_G L(G, D^*) = \int_{x=-\infty}^{x=\infty} \left(p_r(x) \log\left(\frac{p_r(x)}{p_r(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_r(x) + p_g(x)}\right) \right) dx$$

עכשיו נחזור להוכיח את הטענה שצינו בתחילת הפרק:

נראה כי כאשר הדיסקרימינטור הגיע לאימון אופטימלי (מסווגת נכונה עבור כל קלט), הערך האופטימלי לפונקציית ההפסד עבור הגנרטור מיוצג על ידי הפונקציה

$$\argmin_G L(G, D^*) = D_{js}(P_r || P_g) - 2 \log 2$$

נפתח את Jensen-Shannon divergence

$$D_{js}(P_r || P_g) = \frac{1}{2} D_{kl}\left(P_r || \frac{P_r + P_g}{2}\right) + \frac{1}{2} D_{kl}\left(P_g || \frac{P_r + P_g}{2}\right)$$

$$\begin{aligned}
&= \frac{1}{2} \int_{x=-\infty}^{x=\infty} \left(P_r(x) \log \left(\frac{P_r(x)}{\frac{P_r(x) + P_g(x)}{2}} \right) \right) dx + \frac{1}{2} \int_{x=-\infty}^{x=\infty} \left(P_g(x) \log \left(\frac{P_g(x)}{\frac{P_r(x) + P_g(x)}{2}} \right) \right) dx \\
&= \frac{1}{2} \int_{x=-\infty}^{x=\infty} P_r(x) \left(\frac{2P_r(x)}{P_r(x) + P_g(x)} \right) dx + \frac{1}{2} \int_{x=-\infty}^{x=\infty} P_g(x) \left(\frac{2P_g(x)}{P_r(x) + P_g(x)} \right) dx \\
&= \frac{1}{2} (\log 2 + \int_{x=-\infty}^{x=\infty} P_r(x) \left(\frac{P_r(x)}{P_r(x) + P_g(x)} \right) dx) + \frac{1}{2} (\log 2 \\
&\quad + \int_{x=-\infty}^{x=\infty} P_g(x) \left(\frac{P_g(x)}{P_r(x) + P_g(x)} \right) dx) \\
&= \frac{1}{2} \left(\log 4 + \operatorname{argmin}_G L(G, D^*) \right)
\end{aligned}$$

כלומר קיבלנו כי הערך האופטימלי עבור הגנרטור כאשר הדיסקרימינטור בערך האופטימלי שלו הוא:

$$\operatorname{argmin}_G L(G, D^*) = 2D_{js}(P_r || P_g) - 2\log 2$$

נוכיח את טענת העזר אשר בה השתמשנו למעלה:

הערך האופטימלי y^* עבור הפונקציה $y = a \log(y) + b \log(1 - y)$ הוא $y^* = \frac{a}{a+b}$

הוכחה: נחפש נקודת קיצון y^* בנקודה ב הגזרת y' מתאפסת

$$y' = \frac{a}{y} - \frac{b}{1-y}$$

$$0 = \frac{a}{y^*} - \frac{b}{1-y^*}$$

$$y^* = \frac{a}{a+b}$$

סוף הוכחה.

מהוכחה של טענת העזר קיבלנו כי הערך האופטימלי של הדיסקרימינטור הוא :

$$D^*(x) = \frac{p_r(x)}{p_r(x) + p_g(x)}$$

נשים לב שעבור המקרה בו הגנרטור מצליח לייצר אובייקטים שנראים אמיתיים לחלוטין נקבל כי $p_r(x) = p_g(x)$, כלומר עבור כל התפלגות לאובייקט הנוצר ברשת הגנרטיבית יש התפלגות דומה לאובייקט מהעולם האמיתי. במקרה שכזה, גם לדסקרימנטור האופטימלי לא תהיה אפשרות להבחין האם האובייקט הגיע מהעולם האמיתי או תוצר של הרשת הגנרטיבית ונצפה שנקבל הסתברות

$D(x) = 0.5$. נשים לב שתוצאה זו תואמת את התוצאה שקיבלנו.

$$D * (x) = \frac{p_r(x)}{p_r(x) + p_g(x)} = 0.5$$

2.2 אימון הרשתות:

אימון הארכיטקטורה מתחיל כאשר שתי הרשתות מאותחלות במשקולות אקראיים. אימון הרשתות צריך להתבצע באופן כזה ששתי הרשתות משתפרות סימולטנית. את העקרון הזה משיגים בכך שכל איטרציה של אימון מאמנים לסירוגין את הרשת הדסקרימנטטיבית והרשת הגנרטיבית.

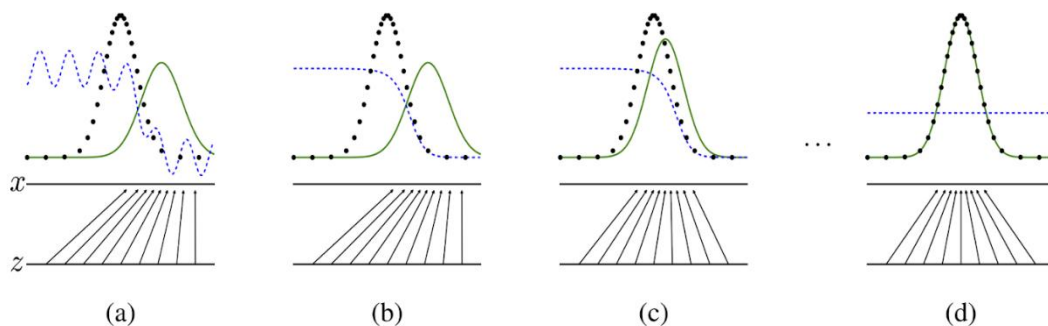
הסיבה לכך שהרשתות צריכות להשתפר סימולטנית היא השימוש באלגוריתם gradient descent לאימון, אשר רגיש למצב של העלמות גרדיינט שקורה במידה שאחת הרשתות מאומנת בצורה ניכרת מהשנייה. במקרה של העלמות הגרדיינט הרשת הגנרטיבית לא תוכל לקבל משוב לאיזה כיוון לשנות את המשקולות על מנת לגרום לאובייקטים שהיא מייצרת לקבל הסתברות גבוהה יותר ברשת הדסקרימנטטיבית, לכן הרשת הגנרטיבית לא תוכל להתקדם ולהשתפר על התוצרים שלה, מכאן שהאימון ייפסק.

באיור 2 ניתן לראות דוגמה אופיינית להתפלגויות של התוצרים של הרשתות לאורך האימון המשותף אשר מצליח להגיע לנקודה האופטימלית עבור שתי הרשתות.

הקו הירוק מייצג את ההתפלגות של פיצ'רים X של האובייקטים הנוצרים מווקטור הרעש z כאשר החיצים מ z ל x מהווים את המיפוי של הרשת הגנרטיבית. בקו הכחול המנוקד, ניתן לראות את ההסתברויות של הערכים המתקבלים על ידי הרשת הדסקרימנטטיבית. בגרף השחור המנוקד, ניתן לראות את התפלגות הערכים של האובייקטים האמיתיים.

בתמונה a, בתחילת האימון הרשתות מאותחלות במשקולות אקראיים ולכן ניתן לראות כי הרשת הגנרטיבית מייצרת אובייקטים בהתפלגות רחוקה מההתפלגות המקורית, והרשת הדסקרימנטטיבית מחזירה ערכים בהסתברות כלשהי. בתמונה b ככל שהאימון מתקדם הרשת הדסקרימנטטיבית יודעת לסווג טוב יותר בין אובייקטים מהעולם האמיתי לרשת הגנרטיבית, ולכן מתקבל מידע יותר אינפורמטיבי לרשת הגנרטיבית. המידע המתקבל הוא עבור איזה מבין האובייקטים שהרשת הגנרטיבית מייצרת דומה יותר לאובייקטים מהעולם האמיתי. המידע מגיע בשלב אימון הרשת

הגנרטיבית בכך שמאמנים את הרשת שייתקבלו ערכים גבוהים ככל הניתן ברשת הדסקרימנטיבית על אובייקטים שהרשת הגנרטיבית מייצרת. בתמונה c הרשת הגנרטיבית מתחילה להשתפר והתפלגות התוצרים שלה מתחילה להתקרב להתפלגות האובייקטים מהעולם האמיתי. בתמונה d, הרשת הגנרטיבית מחזירה אובייקטים אשר נראים כאילו הם אובייקטים מהעולם האמיתי והרשת הדסקרימנטיבית מחזירה 0.5 לכל איבר מכיוון שלא ניתן להפריד בין P_r, P_g .



איור 2: ההתפלגויות של תוצרי הרשתות בארכיטקטורה, והצגת השינוי עם התקדמות האימון. התמונה למעלה נלקחה מ [1], התמונה מתארת שינוי אופייני של תוצרי הרשת עבור אימון של רשתות יריבות.

נקודה שחשוב להדגיש באימון, מכיוון שלרשתות יש פונקציית מטרה שונה זו מזו, בכל שלב כאשר מאמנים רשת אחת בלבד, מאפשרים שינוי במשקולות רק של רשת זו לכיוון האופטימיזציה של פונקציית ההפסד שרשת זו רוצה לטייב.

בשלב האימון של הרשת הדסקרימנטיבית, מכניסים אליה קלטים משני מקורות, האחד אובייקטים שנדגמו מהעולם האמיתי, והשני אובייקטים שנוצרים על ידי הרשת הגנרטיבית על ידי בחירה של וקטורי רעש. כיוון שינוי המשקולות במקרה זה יהיה להגדיל את פונקציית השגיאה ולכן נשתמש ב gradient ascent עבור מיקסום השגיאה בפונקציה:

$$\arg\max_D L(G, D) = E_{x \sim P_r}(\log(D(x))) + E_{z \sim P_z}(\log(1 - D(G(z))))$$

בשלב אימון הרשת הגנרטיבית מעבירים לרשת הדסקרימנטיבית רק תמונות שיוצרו מהרשת הגנרטיבית על ידי בחירה של וקטורי רעש באקראי, כיוון שינוי המשקולות יהיה למזער את פונקציית השגיאה ולכן נשתמש ב gradient descent. נשים לב שמכיוון שהחלק השמאלי בפונקציית ההפסד המקורית לא מכיל איבר התלוי ברשת הגנרטיבית ההשפעה בשלב זה היא רק על החלק של הדוגמאות שהם לא מהעולם האמיתי, החלק הימני.

רשת הגנרטור רוצה למזער את פונקציית המטרה הבאה על ידי שינוי משקולות הגנרטור בכיוון gradient descent:

$$\operatorname{argmin}_G L(G, D) = E_{z \sim p_Z} (\log (1 - D(G(z))))$$

אלגוריתם 1 – אימון רשת GAN
<p>K – מספר צעדים לאימון הדיסקרימנטור</p> <p>m – כמות האובייקטים הנדגמים בכל שלב</p> <p>N – מספר צעדי אימון לרשת הGAN</p> <p>K – מספר סבבי אימון של רשת דסקרימנטור עבור סבב אימון יחיד של רשת גנרטור</p> <p>θ_g – משקולות רשת הגנרטור</p> <p>θ_d – משקולות רשת הדיסקרימנטור</p>
<p>1. בצע N צעדים:</p> <p>1.1. בצע K צעדי אימון לרשת הדיסקרימנטור:</p> <p>1.1.1. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.1.2. דגום בהתפלגות p_r, m אובייקטים מבסיס הנתונים $\{x^1, x^2, x^3, \dots, x^m\}$</p> <p>1.1.3. עבור $i=1 \dots m$ בצע:</p> <p>עדכן את משקולות הדיסקרימנטור לפי כיוון עולה בגרדיאנט בשיטת SGD על פונקציית המטרה:</p> $\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \log (D(x^i)) + (\log (1 - D(G(z^i))))$ <p>1.2. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.3. עדכן את משקולות הגנרטור לפי כיוון יורד בגרדיאנט בשיטת SGD על פונקציית המטרה:</p> $\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m (\log (1 - D(G(z^i))))$ <p>SGD* - stochastic gradient descent, במאמר השתמשו ב SGD עם מומנטום, אלגוריתם עדכון משקולות המנסה להתגבר על מהמורות קטנות בגרדיאנט של פונקציית ההפסד ובכך להימנע מלהיתקע במקסימום מקומי.</p>

מכיוון שבתחילת האימון, הדגימות המיוצרות על ידי הגנרטור לא דומות לחלוטין לאלו מסט האימון, אנו מתחילים מאימון איטי, הסיבה לכך היא שהביטוי $D(G(z^i))$ יקבל ערכים קרובים ל 0 דבר הגורם לכך שהגרדיאנט של הגנרטור יהיה קטן. לכן, במקום לחפש מינימום של הביטוי

$E_{z \sim p_z}(\log(1 - D(G(z))))$ נהוג לבצע שינוי קטן על פונקציית המטרה של הגנרטור, כך שעכשיו ננסה למקסם את הביטוי $E_{z \sim p_z}(\log(D(G(z))))$. הביטויים לא שווים לגמרי אך שניהם מובילים לאותו פתרון של בעיית אופטימיזציה אותה הם מייצגים. הסיבה לשינוי נעשית מכיון שהביטוי החדש עובד יותר טוב נומרית ומצליח לשפר את הגנרטור בצורה יעילה יותר בתחילת האימון. כמו בכל אלגוריתם של למידת מכונה גם לרשתות GAN ישנה נטייה לבעיית הטיה, אשר יכולה לנבוע מצורת האימון, או מדאטה מוטת ולא מגוון. ניקח לדוגמה רשת שמאומנת לייצר פרצופים עם דאטה של תמונות אמיתיות המכיל רק פרצופים של אנשים עם גוון עור מסוים לא נוכל לצפות שרשת GAN תייצר לנו הרבה פרצופים עם גוון עור אחר.

נציין שלשם יכולת שליטה על רמת האמינות Fidelity וגיוון diversity ניתן לאמן את רשת ה-GAN עם וקטור רעש הנדגם עם התפלגות ידועה מראש, כך שלאחר האימון נוכל לבקש לקבל תמונה מרשת ה-GAN על ידי בחירת היפרפרמטרים שישלטו על בחירת וקטור הרעש מאיזה אזורים הוא ידגם, אזורים שיידיגמו מהתפלגות ווקטור הרעש בהסתברות גבוהה יאומנו יותר ולכן ה-Fidelity יהיה גבוה יותר, אם נאפשר יותר אזורים מההתפלגות נקבל יותר גיוון לווקטור הרעש ולכן גם ה-diversity יגדל.

2.3 בעיות נפוצות הנוצרות במהלך אימון רשת GAN

למרות שרשתות GAN מראות תוצאות מרשימות בשנים האחרונות במשימות שנחשבו קשות ליישום ובתוצרי תמונות חדות, במקטע זה בעבודה נראה כי בפועל קשה לאמן את הרשת עד להגעה לנקודה האופטימלית כפי שמתוארת במקטע 2.1.1 בעבודה זו. בהרבה מקרים ארכיטקטורות GAN מסיימות אימון כאשר רשתות הגנרטור יודעות לייצר מספר מצומצם של מחלקות (שאוילי נראים טוב מכיוון שהאימון ממשיך רק על מספר מצומצם של מחלקות אלו). בפועל, זו תוצאה לא רצויה כי היא לא מייצגת את התפלגות האובייקטים מהעולם האמיתי. מקרה נוסף למקור של בעיה באימון הוא כאשר רשת אחת מצליחה לעשות את העבודה שלה בצורה טובה מידי יחסית לרשת השנייה ובכך לגרום לאימון להיעצר בגלל בעיית העלמות הגרדיאנט. מקור נוסף לקושי באימון רשתות GAN הוא בכך שכל רשת בארכיטקטורה בתורה מנסה לשפר את התוצאה הנגדית של הרשת השנייה, ובכך יכולות להיווצר אופטימיזציות מקומיות עבור אובייקטים או תנודות באימון שייגרמו לאימון לא להתכנס לנקודה האופטימלית.

2.3.1 בעיית המימדים הנמוכים

כאשר נחשוב על התפלגות בסיס הנתונים של תמונות המקור שלנו P_r , נבין כי ההתפלגות שוכנת במימד בעל אילוצים חסומים הנובעים מהאילוצים פיזיקליים של העולם האמיתי. למשל אם נחשוב על תמונות של כלבים או חתולים נקבל כי לתמונה צריך להיות מספר רגלים מוגדר ומספר עיניים מוגדר, אם נחשוב על תמונות של גורדי שחקים נבין כי המבנים צריכים להיות ארוכים וישרים. אילוצים אלה

גורמים למרחב בסיס נתוני המקור להיות ממרחב קטן יחסית למרחב הנפרש מכל אפשרויות של הפיקסלים בתמונה.

מרשת הגנרטור אנחנו רוצים שתיווצר לנו תמונות הממופות מווקטור הרעש אשר גודלו קטן מכמות האפשרויות של פיקסלים בתמונה בצורה משמעותית.

מכיוון שמדובר במשטחים בעלי ממד נמוך הנמצאים בתוך מרחב בעל ממד גבוה קיימת סבירות גבוהה שלא יהיה שום חיתוך בין המשטחים. המצב הנפוץ שקורה בתחילת האימונים של הרשת הוא שמשקולות הגנרטור הן אקראיות ולכן לא נצפה שהגנרטור ייצר איברים הנראים אם אילוצים מהעולם האמיתי מהתפלגות P_r , כלומר נקבל כי P_r, P_g יהיו בהסתברות גבוהה שונות זו מזו ולא באותם מרחבים אפשריים.

כאשר אין חפיפה בין המרחבים P_r, P_g , ניתן למצוא דסקרימנטור אופטימלי. מהטענה שהראנו בסעיף 2.1.1 כאשר פונקציית הדיסקרימנטור מגיעה לערכים האופטימליים פונקציית ההפסד האופטימלית עבור הגנרטור היא

$$\argmin_G L(G, D^*) = D_{js}(P_r || P_g) - 2 \log 2$$

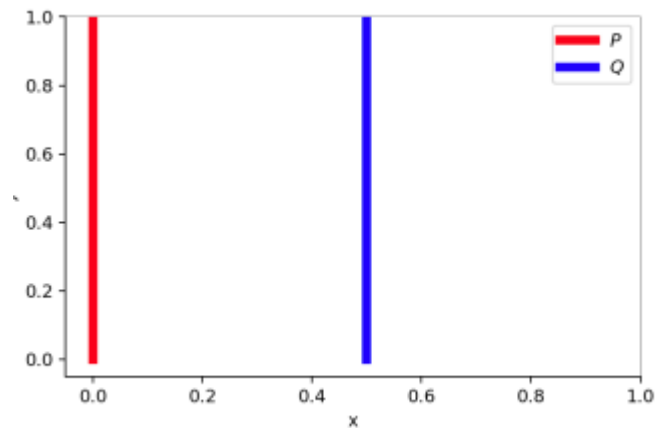
ערכי $D_{js}(P_r || P_g)$ קבועים ככל שההתפלגויות P_r, P_g לא יושבות על אותם מרחבים. במצב שכזה פונקציית ההפסד עבור הגנרטור תהיה קבועה ולכן הגרדיאנטים שלה אפסיים כלומר עבור עדכוני המשקולות באימון של G , כל עוד ההתפלגויות לא יהיו קרובות עדכון משקולות הגנרטור בעזרת אלגוריתם gradient descent יהיה לא אפקטיבי.

בהמשך עבודה זו נראה פונקציות הפסד אשר משתמשות במטריקות מרחק עבור התפלגויות, העדיפות על פני Jensen–Shannon divergence, אחת מהן היא Earth Movers distance (EM). במאמר [3] המחברים מראים דוגמה מייצגת לעדיפות שימוש ב Earth Movers distance, למדידת מרחק בין התפלגויות. בדוגמה ניתנות שתי התפלגויות מהמרחב הדו מימדי אשר כל אחת נמצאת בתת מרחב אחר.

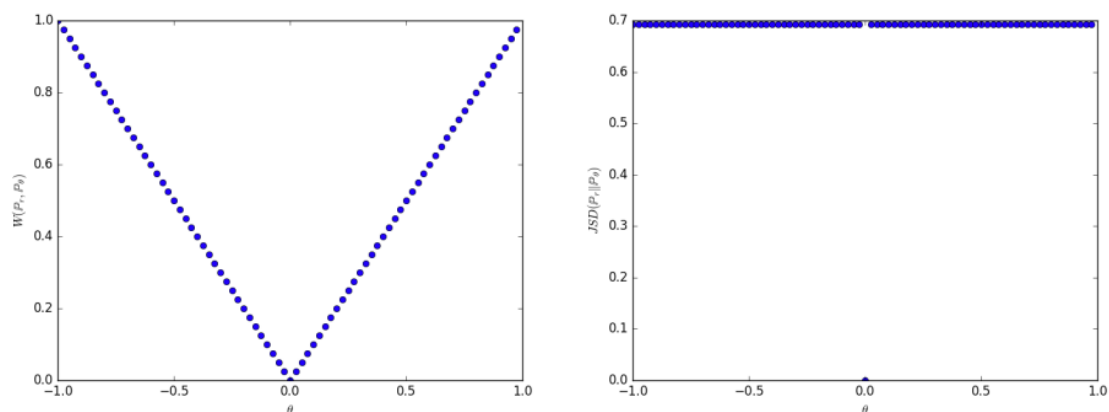
בדוגמה ניתן לראות שבשימוש במטריקות Jensen–Shannon divergence נקבל מרחק בין ההתפלגויות שהוא קבוע למרות שההתפלגויות נהיות קרובות יותר ויותר אחת לשנייה. עוד ניתן לראות שמכיוון שכך, הגרדיאנט על מרחק שכזה מתאפס. זאת בניגוד למרחק earth movers אשר בשימוש בו יש משמעות למרחק בין ההתפלגויות תמיד והגרדיאנט רציף למקוטעין בכל התחום.

הדוגמה המובאת במאמר [3] מוצגת באיור 3. מגדירים שתי התפלגויות Q, P במרחב הדו מימדי. התפלגות P אשר משתנה x שלה עובר בראשית הצירים ומשתנה y שלה הוא משתנה יוניפורמי $U(0,1)$. התפלגות Q מוגדרת בעזרת משתנה x הנשלט על ידי פרמטר θ ובעזרת משתנה יוניפורמי $y \sim U(0,1)$. ככל שפרמטר θ יהיה קרוב ל 0 כך ההתפלגויות יהיו קרובות אחת לשנייה.

באיור 4 ניתן לראות את תוצאות חישוב המטריקות עבור הדוגמה בשינוי פרמטר θ , בצד ימין מרחק Jensen-Shannon ובצד שמאל מרחק Earth Movers. ניתן לראות שמרחק JS לרוב לא אינפורמטיבי לגבי המרחק בין ההתפלגויות P, Q ונשאר קבוע ולכן הגרדיאנט שלו נשאר קבוע לרוב ובנוסף לא רציף. ניתן לראות שמרחק EM נשאר אינפורמטיבי לכל אורך המרחב והגרדיאנט שלו נשאר רציף למקוטעין.



איור 3: דוגמה להתפלגויות אשר בעזרתן במאמר [3] מיוצג ההבדל באינפורמציה למדידת מרחק התפלגויות עבור מטריקות Jensen Shannon ו earth movers. ההתפלגויות הן התפלגויות עבור משתנים הנמצאים במרחב דו מימדי, מימד ראשון ציר X ובו ההתפלגות P נמצאת על ערך 0 והתפלגות Q נמצאת על ערך הנשלט על ידי פרמטר θ , מימד שני הוא ציר Y ועבור שתי המשתנים הערך מוגרל באופן יוניפורמי בתחום בין 0 ל 1. התמונה נלקחה מ [2].



איור 4: תוצאת חישובי המטריקות לחישובי מרחק בין ההתפלגויות התפלגויות P, Q המתוארים באיור 3 מימין מטריקת JS, משמאל מטריקת EM. ציר X מתאר את ערך הפרמטר θ , ציר y מתאר את תוצאת חישוב המטריקה הדוגמה נלקחה מ [3]

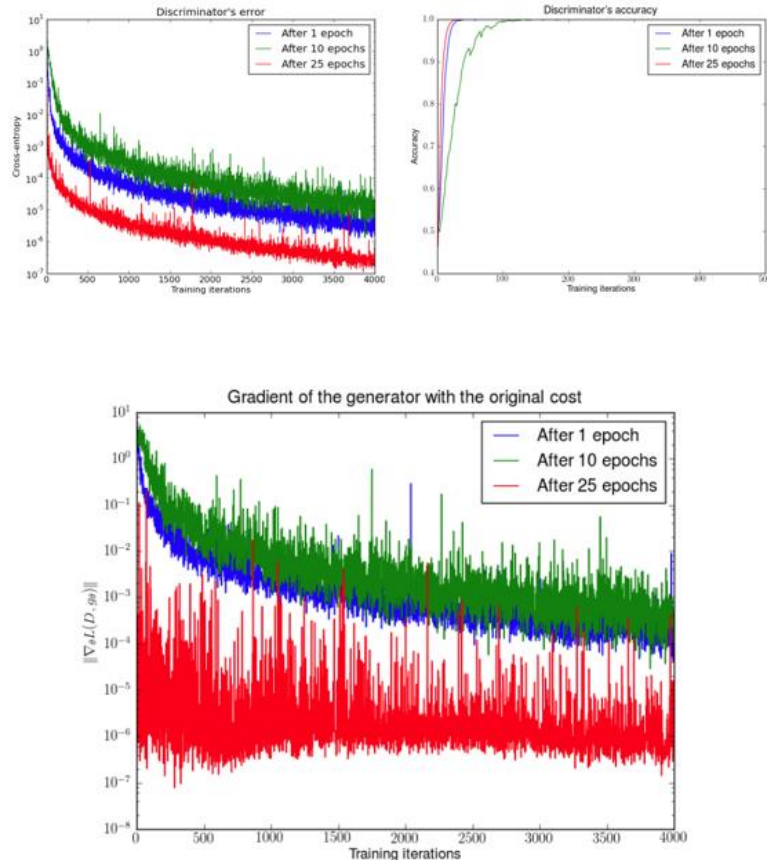
2.3.2 בעיית העלמות הגרדיאנט

הדיסקרימיננטור משתפר ככל שאימון הרשת ממשיך. באופן עקרוני היינו רוצים כי הדיסקרימיננטור ידע להפריד בין אובייקטים שהגיעו מהעולם האמיתי או אובייקטים שייווצרו מהרשת הגנרטיבית. הסיבה לכך היא שהרשת הגנרטיבית משתמשת במשוב של הרשת הדסקרימינטיבית לדעת איזה איברים נראים כאילו הגיעו מהעולם האמיתי בהסתברות גבוהה, על מנת שתוכל לייצר אובייקטים שכאלה בהמשך.

הבעיה שקורית ככל שרשת הדיסקרימיננטור טובה יותר ביחס לרשת הגנרטור היא כאשר הרשת הדסקרימינטיבית מאומנת יותר ויודעת להפריד בהסתברות גבוהה האם אובייקט הגיע מהעולם האמיתי או נוצר מהרשת הגנרטיבית ערכי $D(G(z^i))$ יהיו קרובים ל 0 וערכי $D(x^i)$ קרובים ל 1 ולכן בשלב עדכון המשקולות לא יהיה עדכון משמעותי מכיוון שהגרדיאנטים יהיו קטנים.

אנו רואים סטירה בין הדרישות שלנו מהדיסקרימיננטור, מצד אחד היינו רוצים דסקרימיננטור שיידע להחזיר משוב מדויק עבור הרשת הגנרטיבית ולכן נרצה אותו מאומן בצורה טובה, ומצד שני אם הדיסקרימיננטור מאומן ועושה עבודה טובה מידי אנו מקבלים כי המשוב שלו נהיה פחות משמעותי עבור הגנרטור, דבר שעוצר את התקדמות האימון של הארכיטקטורה.

במאמר [4] מראים דוגמה לאימון לא מאוזן של הארכיטקטורה, אשר גורם להעלמות הגרדיאנט. בתמונות באיור 5 מראים תוצאות לאחר אימון של DCGAN אשר אומן 1,10,25 סבבי אימון. לאחר מכן המחברים קיבעו את אימון הגנרטור, אימנו את הדיסקרימיננטור בלבד, ומדדו את תוצאות פונקציית ההפסד והגרדיאנטים. בתמונה הימנית עליונה ניתן לראות את דיוק הדיסקרימיננטור עולה מהר, בתמונה העליונה שמאלית ניתן לראות את פונקציית ההפסד יורדת בצורה מהירה לערכים הקרובים ל 0, בתמונה התחתונה ניתן לראות את הגרדיאנט של פונקציית ההפסד, אשר מגיע לערכים אפסיים. איור 5 מראה את העבודה שכאשר הדיסקרימיננטור משתפר יותר מידי ביחס לגנרטור, הגנרטור מקבל משוב בגרדיאנט שנעלם ולכן האימון שלו נהיה איטי.



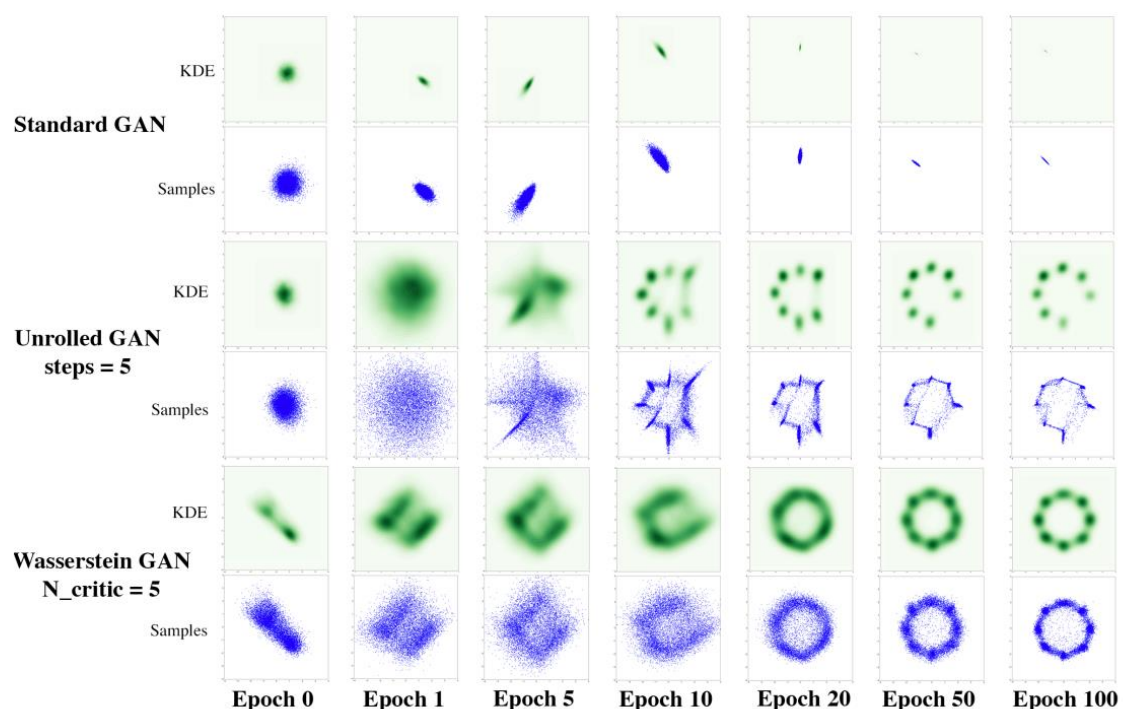
איור 5 : הדגמה של תוצאות המתקבלות מאימון דסקרימנטור לאורך כמה איטרציות, בצורה עדיפה על פני הגנרטור. בתמונות הימנית ניתן לראות את דיוק הדסקרימנטור, בתמונה השמאלית את ערכי פונקציית ההפסד של הדסקרימנטור, בתמונה התחתונה ניתן לראות את ערכי הגרדיאנט של פונקציית ההפסד. האיורים נלקח ממאמר [4].

2.3.3 בעיית קריסת מצב

קריסת מצב (mode collapse) זהו מקרה בו הארכיטקטורה מייצרת מגוון תמונות מצומצם וממשיכה לאמן ולשפר רק את המגוון התמונות המצומצם הזה. הסיבה להגעה לקריסת מצב נעוצה בקטע 1.3 באלגוריתם 1, עדכון המשקולות של הגנרטור $\frac{1}{m} \sum_{i=1}^m (\log(1 - D(G(z^i)))$ אשר נעשה בכיוון שימצער את פונקציית ההפסד כלומר יגרום לרשת הדסקרימנטור להחזיר הסתברות גבוהה על $G(z^i)$. במקרה שיש אובייקטים עבורם $D(G(z^i))$ יקבל ערכים גבוהים מאחרים, לרשת הגנרטור יהיה תמריץ להמשיך לייצר אובייקטים אלה. במקרה המתואר גם בשלבים הבאים באימון הרשת תמשיך לייצר אובייקטים מהסוג שהרשת הדסקרימנטבית טעתה עליו בהתחלה, זאת מכיוון שמשקולות הרשת הגנרטיבית כבר התעדכנו בכיוון שייצר את אובייקטים אלה. ייצור האובייקטים מהסוג המסוים ימשיך עד אשר עדכון משקולות הרשת הדסקרימנטבית יגרום לכך שעבור אותם אובייקטים המסוימים שהרשת הגנרטיבית יודעת לייצר, תוצאותיהם ברשת הדסקרימנטבית יתקבלו בהסתברות נמוכה. לאחר עדכון משקולות שכזה, שוב הרשת הגנרטיבית תעדכן משקולות בתורה לכיוון אובייקטים אחרים עבורם $D(G(z^i))$ יקבל ערכים גבוהים מאחרים, אשר יגרמו לרשת הדסקרימנטבית שוב לטעות. ככה

האימון המנוגד של הרשתות יסתובב במעגל בין המחלקות ולא יגיע לנקודה אופטימלית לפיזור ההתפלגויות ביניהם. דרישה מארכיטקטורת GAN טובה היא שתייצר אובייקטים הנראים מהעולם האמיתי ובהתפלגות המתאימות להם. בהמשך עבודה זו בפרק 5 נדון בכמה מטריקות המעריכות את איכות תוצאת הארכיטקטורה, אחד המדדים לאיכות כפי שמוצע בסעיפים 5.3.3, 5.3.4 הוא מידת הגיון של התמונות.

באיור 6 ניתן לראות הדגמה לקריסת מצב עבור רשת גנרטיבית רגילה. באיור מוצג אימון של מגוון ארכיטקטורת GAN המאומנות למספר סבבי אימון כאשר בכל תמונה ניתן לראות את תוצרי הרשת הגנרטיבית. האימון נעשה על דוגמת צעצוע של 8 גאוסיינים המפוזרים במעגל במבנה דו מימדי. בשתי השורות הראשונות ניתן לראות אימון של רשת GAN אשר הגיע לקריסת מצב. המודל מצליח לייצר מספר מצומצם של אובייקטים, כאשר בכל פעם סוג האובייקטים משתנה. בשתי השורות התחתונות ובשתי השורות האמצעיות מוצגות תוצרים של ארכיטקטורות אשר נדבר עליהן בהמשך עבודה זו, ומטרתם היא למנוע את קריסת המצב. מטרת הארכיטקטורה אשר תוצאותיה מוצגות בשתי התמונות האחרונות היא למזער את המרחק של התפלגות האובייקטים מבסיס הנתונים להתפלגות האובייקטים היוצאים מהגנרטור, נתון מעניין שניתן לראות בשתי התמונות התחתונות עבור ארכיטקטורה זו הוא שראשית נלמד מבנה ההתפלגות במימדים הנמוכים ורק לאחר מכן נלמד קירוב יותר טוב של ההתפלגויות הספציפיות בכל גאוסין.



איור 6: הדגמה של התפלגות תוצרי הרשת הגנרטיבית במגוון ארכיטקטורות לאחר מספר סבבי אימון. הדוגמה נלקחה ממאמר [5]. ניתן לראות קריסת מצב באימון רשת GAN רגילה.

3. סקירת ארכיטקטורת ושיטות אימון אשר משפרות את אימון הרשת

כפי שתיארנו במקטע 2.3 בעבודה זו, אימון ארכיטקטורת GAN הוא משימה קשה. בשל הבעיות המתוארות והרצון להשתמש בתוצרי GAN, שיפור יציבות האימון של רשתות GAN הוא תחום פעיל מאוד מבחינה מחקרית. במקטע זה נתאר כמה ארכיטקטורת אשר מטרתן היא לשפר תוצרי רשת ה-GAN. לצורך שיפור באימון רשתות GAN משתמשים במגוון שיטות הכוללות שינויי במספר עקרונות אפשריים כדוגמאת שינוי בצורת האימון, שינויי בארכיטקטורה, ושינויי בפונקציית ההפסד.

3.1 unrolled generative adversarial networks

במאמר [5] המחברים מתארים שיטה אשר בעזרתה הגנרטור מתאמן לפי דסקרימנטור אופטימלי עבור הנקודה הנוכחית בה הגנרטור נמצא. בשיטה זו המחברים מצליחים לפתור את הבעיה המתוארת במקטע 2.3 בעבודה זו, בה עבור הדיסקרימנטור אנחנו מבקשים שתי מטרות שונות, האחת שהוא יהיה מדויק וידע לתת משוב עבור רשת הגנרטור האם התוצרים שלה נראים אמיתיים, והשנייה שיהיה בעל ערך במושג המוחזר בעדכון המשקולות לפי הגרדיאנטים לרשת הגנרטור, כלומר שלא יגיע לנקודה האופטימלית שלו ובה הוא מחזיר ערכים קרובים לערך האופטימלי הרצוי עבור קבלה או דחיה של איבר. מחברי המאמר מציעים ארכיטקטורה ובה רשת הגנרטור בכל שלב תתאמן עם הדיסקרימנטור מכמה סבבי אימון ובניגוד לכך רשת הדיסקרימנטור תקבל עדכון רק לגבי הגנרטור הקיים בשלב סבב אימון זה. חשוב לשים לב כי שיטה זו שונה מהשיטה שהוצעה במאמר המקורי לרשתות GAN [1], לפי אלגוריתם 1 בשורה 1.1 עבור כל עדכון גנרטור נבצע K עדכוני דסקרימנטור. בשיטה שהוצעה במאמר [1] מבצעים K עדכוני דסקרימנטור עבור כל עדכון גנרטור. בשיטה ממאמר [1] הדיסקרימנטור מקבל עדיפות על פני גנרטור בשלב הנוכחי שלו.

מחברי המאמר מסבירים שהיתרון בשיטת unrolled GAN הוא כמובן בכך שהגנרטור, שלו משימה קשה יותר, יוכל להתאמן יותר מרשת הדיסקרימנטור, שלה משימה קלה יותר. בשיטה זו הגנרטור יוכל לראות את עתיד התפתחות רשת הדיסקרימנטור ולפעול בהתאם להתפתחות זו, ובכך ניתן לרשת הגנרטור יותר כוח בלמידה על פני רשת הדיסקרימנטור אשר רואה רק את תוצרי הגנרטור בשלב הנוכחי שלו. הסיבה שלדסקרימנטור משימה קלה יותר היא מכיוון שהרשת דסקרימנטור צריכה לדעת להחזיר תשובה של ספרה אחת להאם איבר הגיע מהעולם האמיתי, לעומתה רשת הגנרטור בעלת משימה קשה יותר והיא לתפוס את התפלגות האובייקטים מהעולם האמיתי ולייצר אובייקטים מהתפלגות זו.

רעיון נוסף העומד מאחורי תכנון הארכיטקטורה הזו הוא באילוף כיוון האופטימיזציה של פונקציית ההפסד. כזכור, פונקציית ההפסד באופן תאורטי הוגדרה כ $\arg\min_G \arg\max_D L(G, D)$ כלומר, באופן תאורטי היינו רוצים למצוא גנרטור אופטימלי שימזער את התוצאות לפי התשובות של הדיסקרימנטור האופטימלי שמנסה ליצור מקסימום הפרדה בפונקציית ההפסד. נזכיר כי הוכחנו, שכאשר האימון מביא את שתי הרשתות למצב של אופטימליות, הארכיטקטורה מגיעה למצב אשר ברשת הגנרטור נוצרים

תוצרים שלא ניתנים להבחנה האם הגיעו מהעולם האמיתי, ובנוסף שהתפלגות האובייקטים הנוצרים הם באותה התפלגות של אובייקטים מהעולם האמיתי.

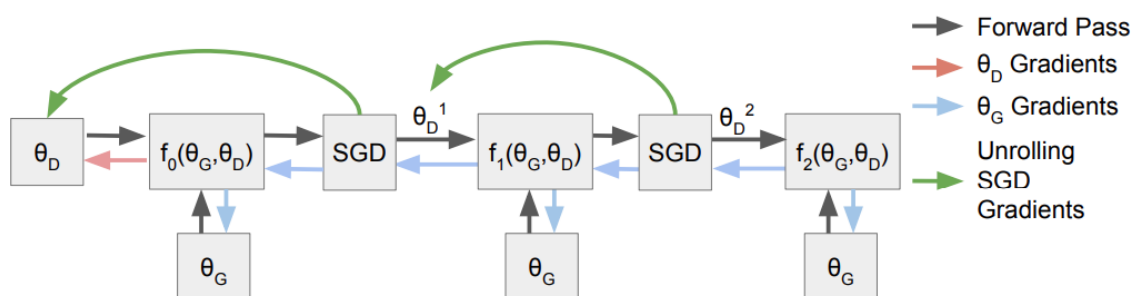
מכיוון שאנחנו מבצעים באיטרציות האימון את שינוי המשקולות לכל רשת בנפרד לפי התוצאות של מצב הרשת הנגדית הנוכחית, אין לנו בטחון שהדבר אכן קורה. מה שיכול להתרחש בשלב מסוים הוא היפוך המשוואה לצורה של $\arg\max_D \arg\min_G L(G, D)$ במקרה שכזה נגיע למצב שראשית הגנרטור יגיע לאופטימליות שלו עבור דסקרימנטור במצב נתון. לכן הגנרטור ייטה את התשובות שלו למקומות בהם תוצאות הדסקרימנטור במצב הנתון ייתנו את הערך להסתברות הכי גבוה. כאשר הדבר קורה עוד לפני שהדסקרימנטור יגיע לאופטימליות שלו, נקבל תוצרים ברשת הגנרטור שלא משקפים את ההתפלגות של האובייקטים מהעולם האמיתי.

בכך שייצרנו אימון שבו הגנרטור רואה את הדסקרימנטור לאחר K אימונים, כאשר K שואף לאינסוף הגנרטור מתאמן על דסקרימנטור אופטימלי ובכך אנו מאלצים אימון של $\arg\min_G \arg\max_D L(G, D)$.

על מנת להגיע לדסקרימנטור אופטימלי נדרשים סבבים רבים של עדכוני משקולות. מחברי המאמר הראו שעבור אימון רשת גנרטור על ארכיטקטורה שנותנת לדסקרימנטור להתאמן לכל היותר 10 צעדים בכל איטרציה, הם מצליחים להוריד את קריסות המצב באופן משמעותי.

נשים לב שעבור עדכון של $K=1$ סבבים הארכיטקטורה מתלכדת עם הארכיטקטורה המקורית של רשת הGAN.

ניתן לראות דוגמה לעדיפות האימון בארכיטקטורה שכזו על פני הארכיטקטורה באימון רגיל באיור 6, בשורות האמצעיות מוצג תוצאות של אימון unroll gun ושתי השורות העליונות מוצגות תוצאות של אימון GAN רגיל.



איור 7: שימוש בארכיטקטורת UNROLLED GAN, דסקרימנטור המתאמן שלושה צעדים וגנרטור המתאמן צעד אחד. באיור רואים בחצים הכחולים את שלב עדכון הגרדיאנטים עבור הגנרטור אשר נעשה עבור כמה דסקרימנטורים מאומנים בשלבי אימון שונים. בחץ האדום ניתן לראות עדכון דסקרימנטור עבור גנרטור משלב האיטרציה הנוכחי. האיור נלקח ממאמר [5]

WGAN - Wasserstein GAN 3.2

מאמר [3] בא לענות על הבעיות המוסברות בסעיפים 2.3, ולצורך אימון יעיל יותר של רשת הדיסקרימינטור מציע להחליף את פונקציית ההפסד של הרשת לפונקציית Wasserstein loss, המחשבת מרחק בין ההתפלגויות P_r, P_g בעזרת מטריקת Earth movers distance.

Wasserstein היא מטריקה המוגדרת על ידי :

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \int d(x, y)^p d_\gamma(x, y)^{1/p}$$

Earth movers distance הוא מקרה פרטי של מרחק Wasserstein כאשר $p=1$

החישוב מחפש את פונקציית העברה בעלת המחיר המינימלי של העברת פוקציה מהתפלגות P_r להתפלגות P_g . מחיר פונקציית העברה γ מחושב על ידי המרחק $d(x, y)$ בין כל נקודה שעוברת כפול כמות הערך שעבר $d_\gamma(x, y)$.

היתרון בשימוש במרחק זה הוא שהוא גזיר למקוטעין ולכן נוכל תמיד להתקדם באימון הדיסקרימינטור (שעכשיו שמו הוחלף לcritic, נרחיב על כך בהמשך), כך נוכל לאמן את ה critic עד לאופטימליות שלו. מכיוון שתמיד יהיה לנו ערך לגרדיאנט של פונקציית ההפסד, תמיד נוכל להתקדם עם אימון רשת הגנרטור.

מטרת פונקציית ההפסד המוצעת היא למצוא את הפונקציה אשר תביא במינימום שינוי את התפלגות של מוצא הרשת הגנרטבית P_g להתפלגות האוביקטים מהעולם האמיתי P_r .

נזכיר כי בניגוד לכך, דסקרימינטורים מארכיטקטורת GAN אחרות המשתמשים בפונקציית הפסד BCE, כפי שהוגדרה בסעיף 2.1, הן בעלי תכונת Jensen-Shannon divergence. כפי שהראנו בסעיף 2.3 בעבודה זו, שימוש בפונקציית הפסד שכזו יכול להגיע למצב של העלמות גרדיאנט ככל שהדיסקרימינטור מגיע לנקודה האופטימלית שלו, בנוסף בנקודה האופטימלית עבור הדיסקרימינטור הערך של פונקציית ההפסד חסום על ידי $-2\log 2$.

בחישוב Earth movers distance, על מנת למצוא את הפונקציה המינימלית שתביא למעבר בין ההתפלגויות ניתן להשתמש בדואליות Kantorovich-Rubinstein אשר מגדירה כי לחישוב המטריקה של המרחק $W(P_r, P_g)$ ניתן למצוא פונקציה f אשר תקבל ערך מקסימלי עבור המרחקים

$$E_{x \sim P_r}(f(x)) - E_{x \sim P_g}(f(x))$$

$$W(P_r, P_g) = \sup_{\|f\|_L \leq 1} E_{x \sim P_r}(f(x)) - E_{x \sim P_g}(f(x))$$

מפונקציה f ישנה דרישה להיות Lipschitz-1 רציפה ומשמע שהנורמה של הגרדיאנט שלה הוא 1 בכל מקום.

מאמר [3] דואג לכך שפונקציה f תהיה Lipschitz-1 על ידי שינוי בעדכון משקולות בכל איטרציה, על ידי ביצוע תחימה של הערכים להיות בטווח מצומצם $[-0.01, 0.01]$. פעולה זו גורמת לכך שהמשקולות יהיו מוכלות רק במרחב קטן ובכך ניתן לשמור את f להיות Lipschitz-1. המאמר מציין ששיטה זו של תחימת המשקולות בעייתית מכיוון שקשה להגדיר מה התחום הנכון כפי שנסביר בהמשך בעבודה זו במקטע 3.3.

ניתן לראות שוב את המטרות בין הרשת הדסקרימינטיבית לגנרטיבית כמשחק MINMAX באופן הבא

$$\argmin_G \argmax_f W(P_r, P_g) = E_{x \sim P_r}(f(x)) - E_{z \sim P_z}(f(G(z)))$$

אלגוריתם 2 – אימון רשת WGAN
<p>K – מספר צעדים לאימון הדיסקרימינטור</p> <p>m – כמות האובייקטים הנדגמים בכל שלב</p> <p>N – מספרי צעדי אימון לרשת הGAN</p> <p>w – משקולות רשת הדסקרימינטיבית</p> <p>θ_g – משקולות רשת הגנרטור</p> <p>C – ערך התחימה של המשקולות</p>
<p>1. בצע N צעדים:</p> <p>1.1. בצע K צעדי אימון לרשת הדיסקרימינטור:</p> <p>1.1.1. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.1.2. דגום בהתפלגות p_r, m אובייקטים מבסיס הנתונים $\{x^1, x^2, x^3, \dots, x^m\}$</p> <p>1.1.3. בצע עבור $i = 1 \dots m$</p> <p>עדכן את משקולות הדיסקרימינטור לפי כיוון עולה בגרדיאנט באלגוריתם RMSProp על פונקציית:</p> $\nabla_w \frac{1}{m} \sum_{i=1}^m (f_w(x^i) - f_w(G(z^i)))$ <p>1.1.4. בצע תחימה של משקולות w לטווח (-C, C)</p> <p>1.2. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.3. עדכן את משקולות הגנרטור לפי כיוון יורד בגרדיאנט באלגוריתם RMSProp על פונקציית:</p> $\nabla_{\theta_g} - \frac{1}{m} \sum_{i=1}^m (f_w(G(z^i)))$ <p>*RMSProp - Root Mean Squared Propagation אלגוריתם לעדכון משקולות, המשנה את קצב העדכון בהתבסס על הערכים האחרונים במשקולות עבור חלון זמן</p>

נחدد את השוני בין מטרות פונקציות ההפסד בארכיטקטורת GAN לארכיטקטורת WGAN.

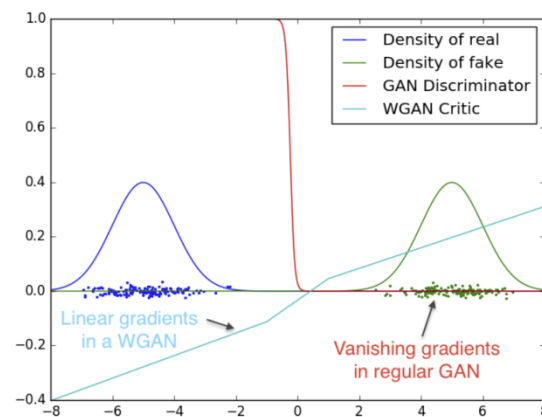
GAN - משתמש ב BCE loss אשר מודד כמה בממוצע הדיסקרימינטור טעה בסיווג שלו על קלטים, כאשר המטרה של הדיסקרימינטור היא למקסם את פונקציית ההפסד. לכן בחלק השמאלי בפונקציית ההפסד בו רק קלטים המגיעים מהעולם האמיתי יכולים להשפיע, הדיסקרימינטור ירצה לתת ערכים בהסתברות גבוהה, ובחלק בו רק קלטים המגיעים מהגנרטור יכולים להשפיע, הדיסקרימינטור ינסה לתת הסתברות נמוכה. כלומר במקרה זה מטרת הדיסקרימינטור היא לדעת לזהות הסתברות להגעת אובייקט מהעולם האמיתי. לצורך ערכים הגיוניים של הסתברות, לרוב משתמשים בסיגמואיד אשר מטיל את ההחלטה לתחום רציף בין 0 ל 1. מטרת הגנרטור במקרה זה היא מנוגדת והוא רוצה לגרום לערך פונקציית הפסד גבוה. הוא יכול לעשות זאת רק בחלק הימני של הפונקציה אשר שם יש לו השפעה, הוא יכול לעשות זאת על ידי ייצור אובייקטים אשר יגרמו לדיסקרימינטור להחזיר תשובה שבהסתברות גבוהה הם הגיעו מהעולם האמיתי.

WGAN – משתמש ב Wasserstein loss אשר מחשב את המרחק Earth movers הפותר את מציאת פונקציית המעבר המינימלית בין שתי התפלגויות P_r, P_g . בארכיטקטורה זו מחברי המאמר החליפו את השם לרשת הדיסקרימינטור בשם קריטיק (critic) מכיוון שמטרת הרשת זו היא ללמוד פונקציה חסומה על ידי תנאי ליפשיץ אשר מודדת את המרחק הגדול ביותר בין שתי ההתפלגויות. זאת בניגוד לרשת הדיסקרימינטור בארכיטקטורה המקורית שמטרתה היתה לסווג האם דוגמה הגיע מהעולם האמיתי או לא. לכן מחברי המאמר מחליפים את שם רשת הדיסקרימינטור בשם קריטיק (critic). במקרה זה הקריטיק מחזיר ערכים שלא חסומים בין 0,1 כמו ב GAN ולכן הוא לא יוצר בעיה של העלמות גרדיאנט ככל שהוא מאומן. מטרת הגנרטור אשר מאומן להקטין את פונקציית ההפסד בארכיטקטורה, משתנה בהתאם לשינוי מטרת רשת הקריטיק, במקרה זה הגנרטור יאומן ליצור אובייקטים שייצרו התפלגות כמה שיותר דומה לאובייקטים מהעולם האמיתי. כפי שניתן לראות באלגוריתם 2 בשורות 1.1.1 – 1.1.4 בעדכון הקריטיק אנחנו מחשבים את מרחק ה Earth movers אשר כפי שהוסבר נותן מידע אינפורמטיבי על המרחק בין ההתפלגויות. עקרון זה גורם לתוצאת השגיאה בפונקציית ההפסד להיות בעלת ערך עם משמעות, ככל שתהיה קטנה יותר נקבל כי ההתפלגויות P_r, P_g קרובות זו לזו. העקרון שערכי השגיאה מפונקציית ההפסד קטנים ככל שההתפלגויות מתקרבות זו לזו הוא חשוב מכיוון שהוא מראה על התקדמות לתוצאה הרצויה. הדבר יכול להועיל בחיפוש אחר רשתות גנרטור. בכך שלפונקציית ההפסד יש ערכים המשקפים את השיפור בהגעה לתוצאה הרצויה, אין צורך בלהסתכל יותר על דגימה של תוצרי תמונות שנוצרו על ידי הרשת הגנרטיביות ולהעריך באיזה רשת יש עדיפות על פני השנייה. כעת ניתן להסתכל על גרפי השגיאה של פונקציית ההפסד ועל ההתקדמות שלהם בלבד. מחברי המאמר מזהירים מלטעות ולהשתמש בשיטת חיפוש זאת גם לרשתות הקריטיק השונות מכיוון שארכיטקטורות שונות של קריטיק יגזרו עדכון גרדיאנט שונה ולכן הגרפים יהיו שונים.

באיור 8 ניתן לראות סיווגים של שתי רשתות הדיסקרימינטור ים השונות, האחד משתמש בפונקציית הפסד BCE והשני משתמש בפונקציית הפסד Wasserstein. הדיסקרימינטור ים אומנו עד לאופטימליות עבור סיווג בין שתי התפלגויות גאוסיות עם תוחלות ושונויות כך שהחפיפה ביניהן לא קיימת כמעט.

לאחר אימון הדיסקרימינטור ים, נמדדו תוצאות סיווג של הדיסקרימינטור ים על ערכים בתחום ההתפלגויות.

ניתן לראות כי עבור אימון הדיסקרימינטור שאומן בעזרת BCE (גרף אדום) כאשר אנחנו באזור ההתפלגות של האיברים מהעולם האמיתי אנחנו מקבלים ערכים שהם 1 ועבור האזור שהם הערכים של האיברים מהעולם המזויף אנחנו מקבלים ערכים שהם 0. כלומר באזורים אלו אין גרדיאנט. ניתן לראות כי עבור הדיסקרימינטור אשר משתמש בפונקציית הפסד Wasserstein (הגרף התכלת) גרדיאנט משמעותי בכל אזור.



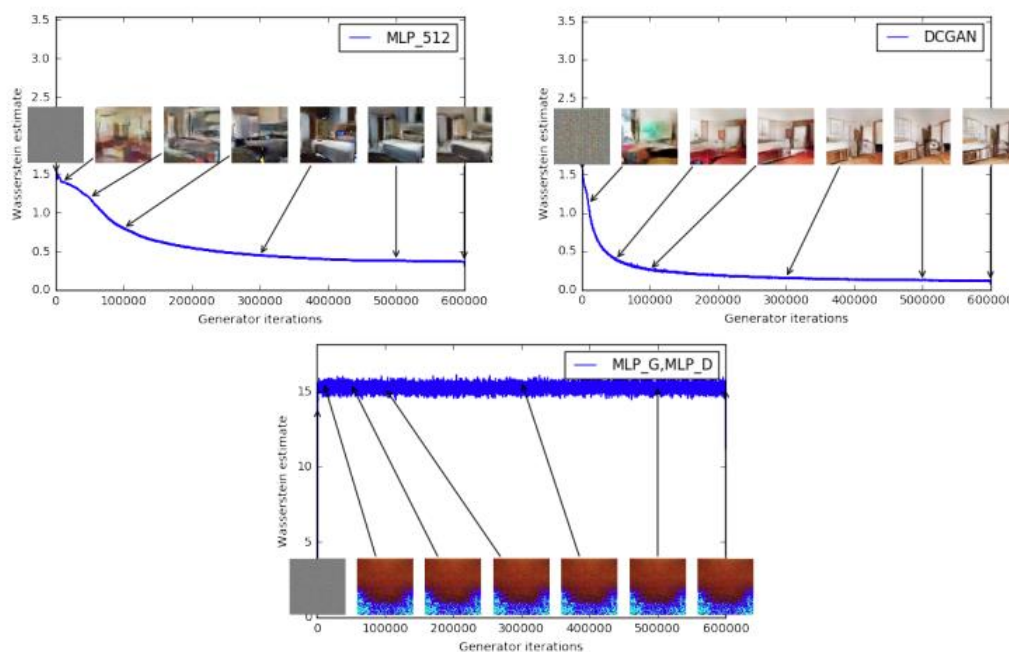
איור 8: תוצאות סיווג של דסקרימינטורים המשתמשים בפונקציות הפסד Wasserstein ו BCE, עבור שתי התפלגויות נורמליות רחוקות זו מזו, ערכי ציר ה X הם הערכים מהתחום של ההתפלגויות, וערכי ציר ה Y אלו התוצאות אשר התקבלו מהעפלת הרשת הדסקרימינטיבית בערכי התחום השונים. התמונה למעלה נלקחה ממאמר [3]

באיורים 10 ו 11 מחברי המאמר מראים את גרף התקדמות השגיאה בפונקציית הפסד עבור מספר ארכיטקטורות אשר בכול איור השתמשו באותה פונקציית הפסד. בפונקציית הפסד Wasserstein (באיור 10) ו BCE (באיור 11). הארכיטקטורות שבהן משתמשים הן Multilayer perceptron עבור הגנרטור בלבד (משמאל למעלה) DCGAN (הימנית עליונה), ו Multilayer perceptron עבור הגנרטור ודסקרימינטור (תחתונה). ההבדל בין איור 10 לאיור 11 מדגיש את הערך והצורך בפונקציית הפסד אשר מתואמת עם ההתקדמות של איכות תוצרי הרשת הגנרטיבית.

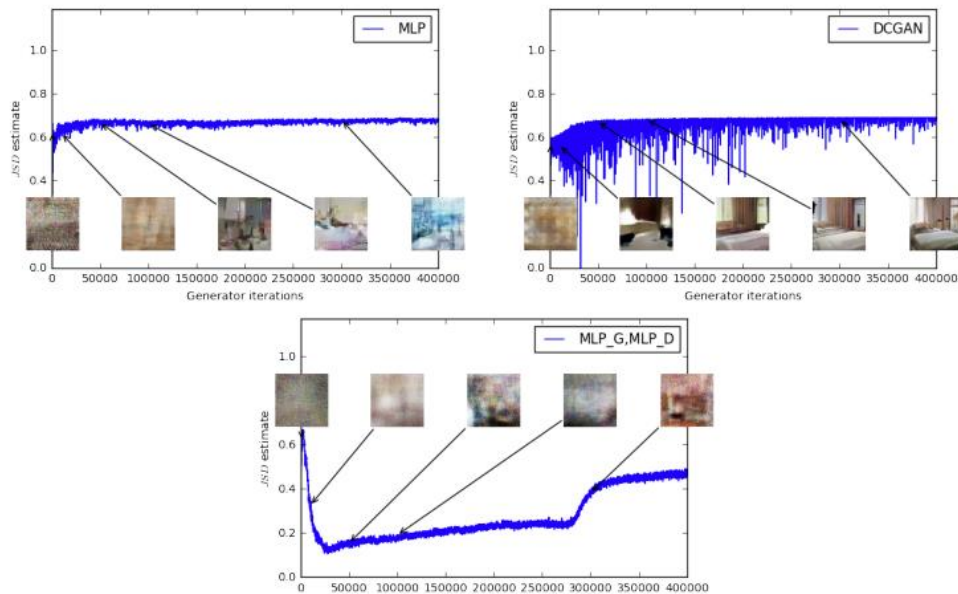
באיור 10 בכל התמונות ניתן לראות שככל שתוצרי הגנרטור טובים גרף ערכי השגיאה של פונקציית ההפסד מתכנס לתוצאות טובות. בגרף התחתון ניתן לראות ארכיטקטורה אשר לא מתכנסת לתוצאה טובה, וכתוצאה מכך ניתן לראות במקרה זה שגרף השגיאה של פונקציית ההפסד לא יורד.

באיור 11 ניתן לראות שגרף השגיאה לא מייצג את התקדמות התוצרים של הרשת הגנרטיבית. בכולן ניתן לראות שעם התקדמות באיכות תוצרי הרשת הגנרטיבית, גרף התקדמות השגיאה לפונקציית ההפסד עולה (במקום לרדת). נשים לב שבגרפים באיורים העליונים שגיאת ההפסד הגיעה לערך המתקרב ל $\log 2$ שזה הערך התחתון עבור הפונקציה

פונקציה השגיאה האופטימלית שהגנרטור מנסה לאפסם לאחר שהדיסקרימינטור הגיע לערך המינמלי בפונקציית השגיאה המשותפת. $\argmin_G L(G, D^*) = D_{js}(P_r || P_g) - 2\log 2$ אשר כפי שהוסבר בסעיף 2.1 בעבודה זו, זוהי

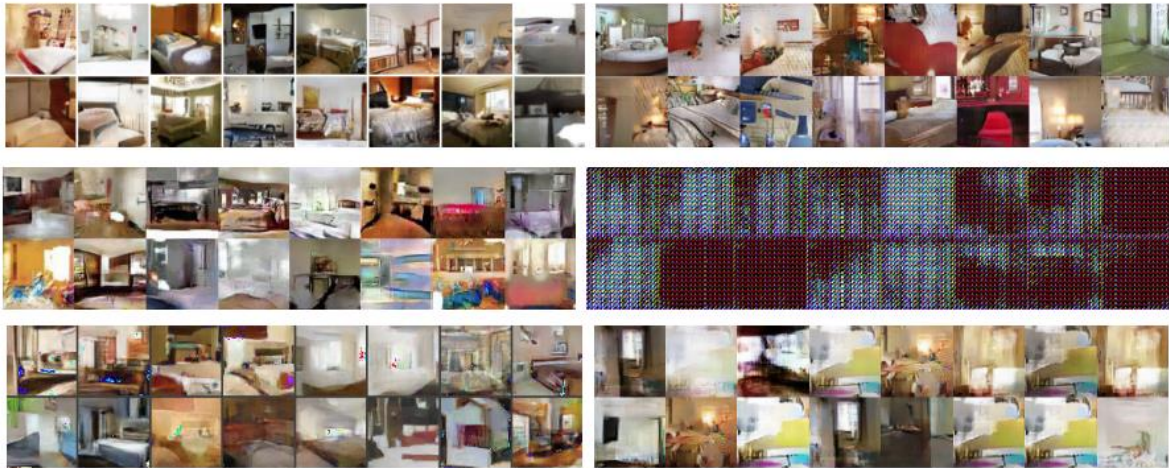


איור 10: ערכי פונקציית ההפסד למול השיפור בתוצרי הרשת הגנרטיבית במגוון ארכיטקטורות אשר משתמשות בפונקציית הפסד Wasserstein



איור 10: ערכי פונקציית ההפסד למול השיפור בתוצרי הרשת הגנרטיבית במגוון ארכיטקטורות אשר משתמשות בפונקציית הפסד BCE

באיור 9 ניתן לראות דוגמה להבדל בין תוצרי הרשת הגנרטיבית כאשר משתמשים בפונקציית הפסד Wasserstein (עמודות משמאל) וכאשר משתמשים בפונקציית הפסד BCE (עמודות מימין). בכל שורת תמונות ניתן לראות שימוש בארכיטקטורת GAN שונה, ניתן לראות שבכל הארכיטקטורות השונות ישנה עדיפות לשימוש בפונקציית הפסד Wasserstein. בשורה הראשונה מובאים תוצרים של אימון רשת DCGAN, שתי הארכיטקטורות הצליחו לייצר תמונה טובה. בשורה השנייה הוצאה מארכיטקטורת DCGAN שכבת ה batch normalization, ניתן לראות שהרשת שאומנה עם פונקציית הפסד Wasserstein נשארה יציבה ואילו הארכיטקטורה שאומנה עם BCE נכשלת לייצר תמונות. בשורה התחתונה ניתן לראות תוצאות של רשת בארכיטקטורה של Multilayer perceptron, ניתן לראות שהרשת שאומנה בפונקציית הפסד Wasserstein נותנת עדיין תוצאות טובות ואילו הרשת שאומנה עם פונקציית הפסד BCE נופלת למקרה של קריסת מצב ומייצרת תוצאות החוזרות על עצמן.



איור 9 : הצגת ההבדלים על מגוון ארכיטקטורות עבור שימוש בפונקציות הפסד BCE (עמודה ימין) ופונקציית הפסד Wasserstein (עמודה שמאל). כפי שמוסבר למעלה עבור כל שורה בוצע שינוי בארכיטקטורה, וניתן לראות כי בכל הניסויים יש עדיפות בשימוש בפונקציית הפסד Wasserstein. התמונה למעלה נלקחה ממאמר [3].

3.3 WGAN-GP- gradient penalty

כותבי מאמר [11] 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 [3]. תחימה קטנה מידי מקשה על אימון מהיר ותחימה גדולה מידי מקשה על שמירה של תנאי ליפשיץ הנדרש באימון. השוני ב WGAN-GP הוא בהוספת תנאי רגולציה לפונקציית ההפסד, אשר עוזר לשמור על תנאי ליפשיץ. בתנאי הרגולציה החדש מודדים את הגרדיאנט של הדסקרימיננטור ביחס לתמונה הנוצרת מאינטרפולציה של שתי תמונות, האחת תמונה הנדגמת מהעולם האמיתי והאחרת תמונה הנוצרת מהרשת הגנרטיבית. ככל שהגרדיאנט עבור התמונה המאוחדת גדול יותר, ככה משקל התוספת של תנאי הרגולציה לפונקציית ההפסד יהיה גדול יותר.

פונקציית ההפסד החדשה היא:

$$\begin{aligned} \argmin_d \argmax_f L(P_r, P_g) \\ = \underbrace{E_{x \sim P_r}(f_w(x)) - E_{x \sim P_g}(f_w(G(z)))}_{\text{original critic loss}} + \underbrace{\lambda E_{\hat{x} \sim P_{\hat{x}}}[(\|\nabla \hat{x} f_w(\hat{x})\| - 1)^2]}_{\text{gradient penalty}} \end{aligned}$$

כאשר החלק השמאלי של הפונקציה הוא פונקציית ההפסד של WGAN והחלק הימני הוא תנאי הרגולציה.

\hat{x} - התמונה הנוצרת מאינטרפולציה של דגימות של תמונה מהעולם האמיתי ותמונה הנוצרת על ידי הגנרטור.

אלגוריתם 3 – אימון רשת WGAN-GP
<p>K – מספר צעדים לאימון הדיסקרימינטור</p> <p>m – כמות האובייקטים הנדגמים בכל שלב</p> <p>N – מספרי צעדי אימון לרשת הGAN</p> <p>θ_g – משקולות רשת הגנרטור</p> <p>θ_d – משקולות רשת הדיסקרימינטור</p>
<p>1. בצע N צעדים:</p> <p>1.1. בצע K צעדי אימון לרשת הדיסקרימינטור:</p> <p>1.1.1. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.1.2. דגום בהתפלגות p_r, m אובייקטים מבסיס הנתונים $\{x^1, x^2, x^3, \dots, x^m\}$</p> <p>1.1.3. דגום בהתפלגות יוניפורמית $U[0,1]$ m איברים $\{\epsilon^1, \epsilon^2, \epsilon^3, \dots, \epsilon^m\}$</p> <p>1.1.4. עבור $i=1 \dots m$ בצע:</p> $\hat{x}^i = \epsilon^i x^i + (1 - \epsilon^i)(G(z^i))$ <p>עדכן את משקולות הדיסקרימינטור לפי כיוון עולה בגרדיאנט באלגוריתם Adam על פונקציית המטרה:</p> $\nabla_w \frac{1}{m} \sum_{i=1}^m (f_w(x^i) - f_w(G(z^i))) + \lambda [(\ \nabla_x f_w(\hat{x}^i)\ - 1)^2]$ <p>1.2. דגום בהתפלגות p_z, m ווקטורי רעש $\{z^1, z^2, z^3, \dots, z^m\}$</p> <p>1.3. עדכן את משקולות הגנרטור לפי כיוון יורד בגרדיאנט באלגוריתם Adam על פונקציית המטרה:</p> $\nabla_{\theta_g} - \frac{1}{m} \sum_{i=1}^m (f(G(z^i)))$ <p>Adaptive Moment Estimation – Adam* אלגוריתם לעדכון משקולות בצורה אדפטיבית, לפי מומנט ראשון ושני של המשקולות.</p>

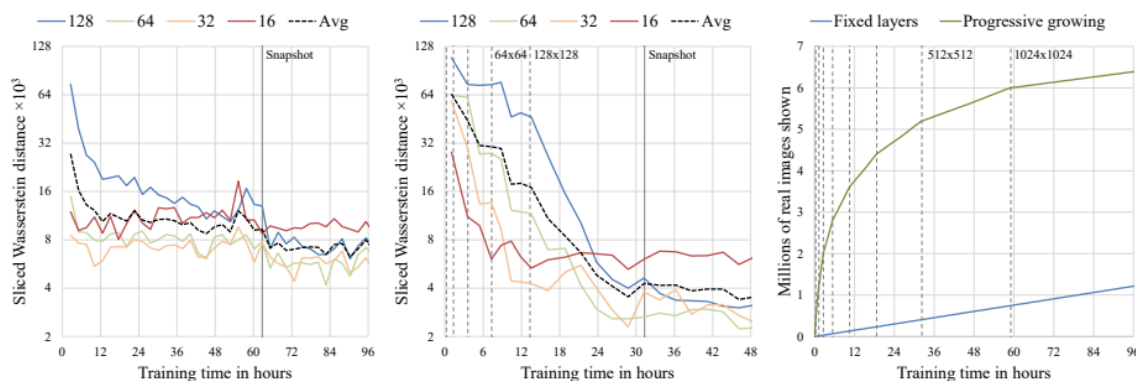
לרוב, ארכיטקטורת המשתמשות במרחק Wasserstein באימון בעזרת רגולציה לגרדיאנט WGAN-GP. בעבודה הארכיטקטורות שמבצעות שימוש באימון שכזה הן SinGan המוצגת בסעיף 4.6 DeblurGani המוצגת בסעיף 4.3.1

3.4 progressive growing of gans

במאמר [23] הראו שימוש בשיטת אימון אשר מאפשרת ליצור תמונות גדולות ברזולוציה גבוהה. שיטת האימון מגדילה את גודל הארכיטקטורה בהדרגה ואת הקלטים ופלטים שהיא עובדת איתם. העקרון העומד מאחורי שיטה זו נועד ראשית לאמן את הארכיטקטורה ללמוד את המבנים הכלליים של

האובייקטים, ולאחר מכן צעד אחר צעד להתקדם ללמוד פרטים קטנים ומדויקים יותר בהתפלגויות בבסיס הנתונים.

המחברים מראים כי התהליך מאיץ את האימון, ומקטין את האפשרות ליצירת אובייקטים הרחוקים מהתפלגות האובייקטים המקוריים. הסיבה לכך היא מכיוון שיותר קל לבצע אימון של דברים בסיסיים ולאט להוסיף עוד מידע ללמידה, כך הארכיטקטורה לומדת בכל שלב רק את המיפוי שהשכבה הנוספת צריכה ללמוד. לימוד בשלבים יותר קל מאשר לבקש ללמוד את כל ההתפלגויות של הפרטים הקטנים של המידע בבסיס הנתונים בבת אחת בכל השכבות. ייתרון נוסף העוזר להאצת האימון, בשלבים המוקדמים באימון הארכיטקטורה מורכבת מרשתות קטנות יחסית, והפעלת רשתות קטנות יותר מאיץ אף הוא את התהליך. באיור מספר 35 ניתן לראות את הניסויים אשר ביצעו המחברים על מנת להוכיח את הטענה שאימון בשיטה שלהם מאיץ את תהליך האימון. בתמונת ניתן לראות קווים מקווקים אשר מתארים שלב של הגדלת שכבה. בגרף מהתמונה הימנית ביותר ניתן לראות את היחס בין מספר התמונות המוצגות לדסקרימנטור ביחס לשעות האימון. כמובן שניתן לראות כי באימון progressive (בירוק) בתחילה היחס גבוה יותר מכיוון שהרשת קטנה יותר, וזמן האימון על פחות שכבות קטן יותר. לאחר שהגענו לגודל המקסימלי, ניתן לראות כי שתי הרשתות עולות באותו שיפוע בגרף. הגרף השמאלי והאמצעי מציגים מרחק sliced Wasserstein distance על כמה שכבות בפרמדיט לפליסיאן. ככל שערכי הגרף נמוכים, כך ההתפלגויות של מאפייני האובייקטים האמיתיים דומים למאפייני האובייקטים המיוצרים על ידי רשת הגנרטור. בגרף השמאלי ניתן לראות אימון של רשת עם ארכיטקטורת GAN רגילה, בגרף האמצעי ניתן לראות אימון של progressive. ניתן לראות כי ערכי הגרפים נמוכים יותר בתמונה האמצעית מהשמאלית בהרבה פחות זמן (30 שעות, למול 60 שעות).

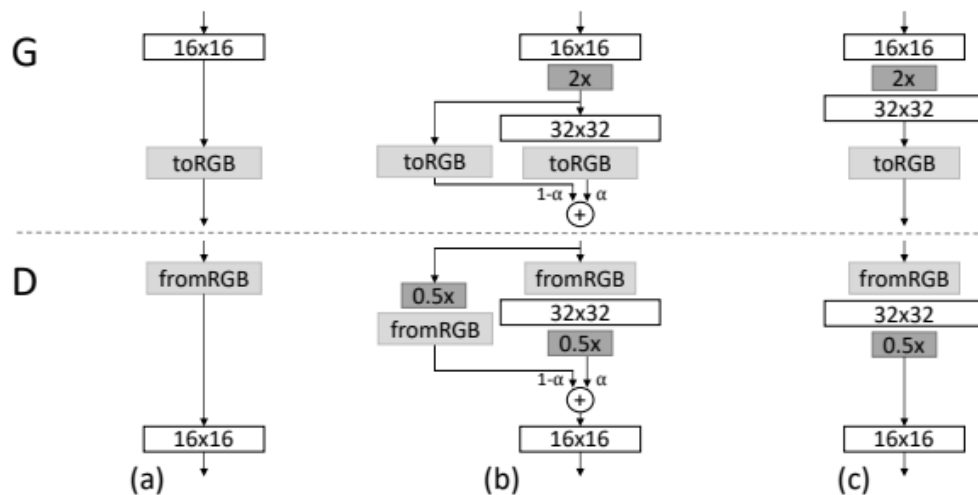


איור 35: בתמונה הימנית, תצוגה גרפית של זמן אימון למול כמות תמונות שרואה הדסקרימנטור. בתמונות האמצעית והשמאלית תצוגה גרפית של מדד sliced Wasserstein distance על מספר שכבות לפליסיאן למול זמן התקדמות האימון. הקווים המקווקים מציגים את הזמנים בהם הארכיטקטורה באימון progressive עברה הגדלה.

מחברי המאמר משתמשים בארכיטקטורת מראה ובה הדיסקרימינאטור והגנרטור זהים והפוכים אחד מהשני בכל שלב, כלומר לשניהם תמיד אותן שכבות המסודרות באופן הפוך. בכל השלבים באימון, כל השכבות הקיימות נשארות לא קפואות והמשקולות שלהן יכולות להשתנות במהלך האימון.

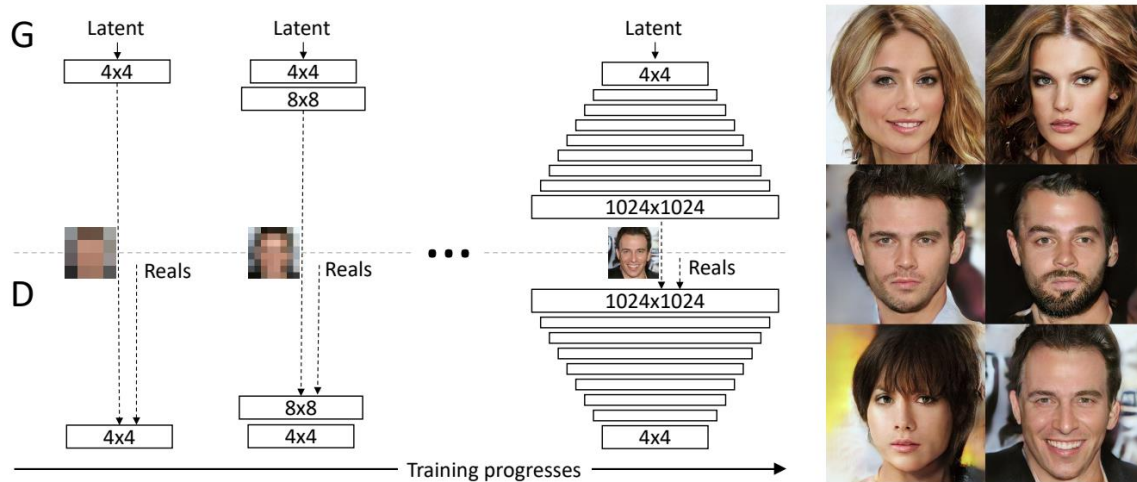
אימון הארכיטקטורה מתבצע בצורה כזו שגדלי התמונות שהיא מייצרת עולה עם הזמן. הדבר נעשה בצורה איטרטיבית. בכל שלב כאשר מגדילים את התוצר של הגנרטור, באופן סימטרי מגדילים את תמונות המקור שרשת הדיסקרימינאטור תקבל. כאשר מוכנסת שכבה חדשה לרשת היא מוכנסת בהדרגה, על מנת לא ליצור השפעה ניכרת משכבה לא מאומנת לשכבה שכבר אומנה על שלב קודם בתהליך.

איור 12 מציג את תהליך הכנסת השכבה הנוספת שנעשה על ידי משקול לכמה מאפשרים לשכבה החדשה להשפיע, המשקול משתנה ככל שהאימון מתקדם. חיבור השכבות נעשה בשיטת חיבור residual אשר מחבר את הכניסה של שכבה אחת למוצא שלה, במקרה זה ביחס הנקבע על ידי פרמטר a הגדל לאורך האימון, מערך 0 עד לערך 1. חיבור residual יוסבר בהרחבה בהמשך עבודה זו בסעיף 4.1.1. ניתן לראות באיור 12 את הוספת השכבות, הדוגמא המצורפת מראה הגדלת הארכיטקטורה מתוצרים של 16×16 כפי שמוצג בשלב a לתוצרים של 32×32 כפי שמוצג בשלב c , במהלך שלב המעבר b ניתן לראות את חיבור ה residual הנשלט על ידי פרמטר a ומאפשר לקבוע כמה השכבה החדשה תשפיע על שינוי המשקולות בשכבה הקודמת באימון.



איור 12: הדגמת תהליך הכפלת הרזולוציה בארכיטקטורה, שלב a הארכיטקטורה עובדת על תמונות בגדלי 16×16 , שלב c הארכיטקטורה עובדת על תמונות בגדלי 32×32 , שלב b שלב המעבר בין גדלי תוצרי הארכיטקטורה, בשלב זה משתמשים בחיבור השכבה הנוספת בחיבור residual אשר נשלט על ידי פרמטר a אשר גדל ככל שהאימון מתקדם. האיור נלקח מ [23]

הדיסקרימינטור מקבל כרגיל תמונות משתי מקורות בשלב האימון, מקור אחד הוא תוצר רשת הגנרטור, ומקור שני הוא תמונות הנדגמות מתמונות המקור העוברת פעולה של הקטנת רזולוציה מתמונת המקור לתמונה בגודל הכניסה המתאימה לדיסקרימינטור בשלב זה.



איור 13: אימון progressive growing. ניתן לראות שככל שמתקדמים באימון (משמאל לימין) נוספות עוד שכבות, גם לגנרטור וגם לדיסקרימינטור באופן של ארכיטקטורת מראה. בכל שלב נכנסות לדיסקרימינטור תמונות מבסיס הנתונים אשר איכות הרזולוציה שלהם גדלה בהתאמה להגדלת הארכיטקטורה. התמונות הנראות בצד ימין אלו תוצרים של הארכיטקטורה בשכבה הסופית אשר עובדת על תמונות בגדלי 1024X1012. האיור נלקח מ [23]

נראה שימוש בעקרון של לימוד בשיטה הדרגתית בסעיף 4.6 בעבודה זו בארכיטקטורת SinGan.

תרומה נוספת של מחברי מאמר [23], לראשונה הוצג כי ניתן לבצע אינטרפולציה לינארית על המרחב הלטנטי ולקבל מעברים חלקים בין תוצרים שונים של הרשת הגנרטיבית. נתאר בהרחבה שימוש מתקדם של טכניקה זו לשינוי ערכי תמונה בסעיף 4.5

4 סקירת ארכיטקטורת ושיטות מובילות לשינוי ערכי תמונה

עד כה התמקדנו ברשתות אשר בקבלת ווקטור רעש בהתפלגות P_z ידעו למפות את תוצרי הרשת לאובייקטים בהתפלגות P_g כאשר המטרה היתה לגרום למודלים להתכנס לנקודה שייצור התפלגות P_g יהיה דומה להתפלגות האובייקטים בעולם האמיתי P_r .

במקטע זה נתאר מגוון ארכיטקטורות אשר בעזרתן ניתן לשלוט על תוצרי התמונה כרצוננו. מטרת ארכיטקטורות אלו היא לייצר תמונה על בסיס מידע המתקבל מווקטור המידע המצורף לכניסה של הדיסקרימינטור ושל הגנרטור.

4.1 שימוש בפונקציות הפסד לשמירה על מבנה התמונה

כפי שנראה במאמרים בהמשך, רוב המאמרים אשר עוסקים בארכיטקטורות למיפוי תמונה ממרחב אחד לאחר כגון צביעה של תמונה, הגדלת תמונה, הורדת ערפל ורעש ועוד, משתמשים בפונקציות

הפסד אשר מודדות את המרחק של תמונת הכניסה X לתמונה הסופית אליה רוצים להגיע Y . פונקציות ההפסד הנפוצות למדידת מרחק בין מיפוי תמונות:

- per pixel loss - מרחק אוקלידי בין פיקסלים – 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. בפועל המרחק מודד תכונה חזקה מידי, למשל 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 מכיוון שמטרתו להגיע למינימום שגיאה של האפשרויות אשר יכול להיות הממוצע של אותן אפשרויות, ולא מחייב ערך אמיתי אפשרי. לרוב מאמרים משתמשים במרחק בנורמה L_2 בין הפיקסלים ולפעמים מאמרים משתמשים במרחק בנורמה L_1 , ניתן לבצע שילוב של כמה מטריקות המודדות מרחק בין פיקסלים מכמה נורמות שונות כפי שנעשה ב [6].

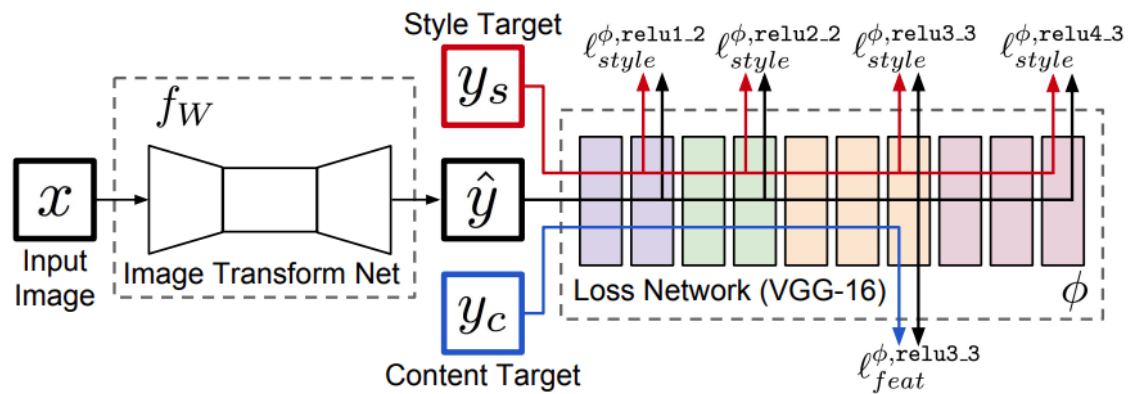
$$L_{L_k}(G) = E_{x,y \sim P_{data}(x,y)} (||Y - G(x)||_k)$$

כאשר k מגדיר את הנורמה בה בוחרים להשתמש.

- Content loss - מרחק מאפייני אובייקטים כפי שקודדו ברשת סיווג בשכבות הפנימיות שלה. כפי שמוסבר ב [9], שימוש בפונקציית הפסד אשר מתייחסת למאפיינים שכאלה משפר משמעותית את איכות תמונת המוצא בכך שמנסים למזער את המרחקים של ייצוג אובייקטים בין תמונת המטרה לתמונת המוצא, דבר העוזר לקרב את מבנה התמונות. לרוב משתמשים ברשת סיווג VGG ובוחרים איזה שכבה רוצים להשתמש להוצאת המאפיינים של התמונה, כאשר $\phi_i(x)$ מייצג את הווקטור של הקונבולוציה בשכבה i (אחרי שכבת אקטיבציה). ניתן לבחור כמה סוגי מרחקים בין פיצרים, לרוב מה שנעשה הוא לקחת המרחק על פי נורמה K בין הפיצרים.

$$L_{Content}(G) = E_{x,y \sim P_{data}(x,y)} (||\phi_i(G(x)) - \phi_i(y)||_k)$$

בתהליך האימון משתמשים ברשת סיווג נוספת לצורך הוצאת ערכי משקולות בשכבה פנימית. הרשת הנוספת נשארת קבועה לאורך כל האימון וערכי המשקולות בה לא משתנים. עבור הרשת אותה רוצים לאמן מאפשרים שינוי במשקולות. בשלב האימון עבור האובייקט הנוצר והאובייקט אליו רוצים להשוות, מבצעים שימוש ברשת הנוספת על מנת לקבל ערכים של שכבה פנימית להשוואה, בערכים אלו מבצעים שימוש לצורך מדידת מרחק בין האובייקטים.



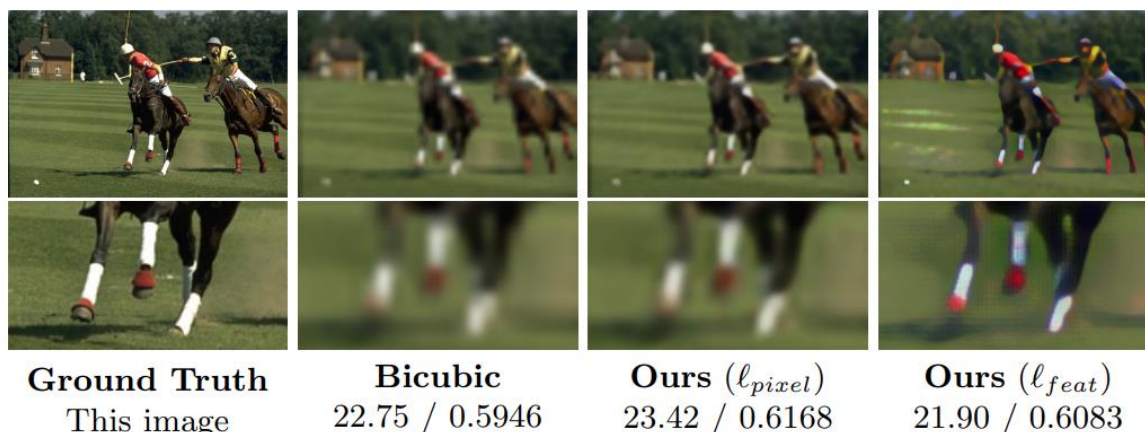
איור 14: שימוש ב-Content loss. התמונה נלקחה מ [9].

נשים לב שבשימוש ב-Content loss בפונקציית ההפסד, ניתן להבחין כי יש שמירה על פרטים יותר מרחביים, כלומר ניתן לראות שמירה על פיצרים יותר הגיוניים, ויותר חדים, ופחות ניסיון לשמירה על הקטנת הטעות הממוצעת של מרחב האפשרויות עבור פיקסל בודד. דבר זה גורם לכך שאנחנו לא בהכרח ננסה לשחזר את האובייקט המקורי אלא אובייקט הנראה עם פיצ'רים דומים לפיצ'רים שנלמדו ברשת הקונבולוציה הנוספת.

בעקרון זה מומלץ להשתמש כאשר אין לנו את המידע המקורי אותו אנחנו רוצים לשחזר, בייחוד כאשר יש מספר אפשרויות להשלמת מידע, ומבנה הפיקסלים המשוחזרים הוא לא ברור. במקרה שכזה נרצה לשחזר מבנה סמנטי נכון של הפיקסלים ולא את המבנה שיביא למינימום שגיאה של האפשרויות אשר יכול להגיע לממוצע של האפשרויות ולכן יהיה מטושטש.

לרוב כאשר נתמקד בשחזור תוכן הפיצרים האפשריים של האובייקטים בתמונה נקבל תוצאות נמוכות במדדים אשר מודדים את המרחק בין הפיקסלים מהתמונה הנוצרת לתמונה שרוצים לשחזר. לכן לרוב מדדים כגון PSNR ו SSIM יקבלו תוצאות נמוכות בשחזור תמונה בעזרת רשתות GAN. המדדים המדוברים והסיבה לתוצאות הלא משקפות את מידת חדות התמונה מוסברים בהרחבה בפרק 5 בעבודה זו.

באיור 15 ניתן לראות המחשה של השיפור שנעשה בהחלפת פונקציית ההפסד עבור אותה ארכיטקטורה, בשימוש בפונקציית הפסד שמחשבת מרחק המשתמש ב-Content loss (תמונה ימנית). ניתן לראות כי הפיצרים המרחביים נשמרים והתמונה מתחדדת. זאת לעומת שימוש ב per pixel loss (תמונה שניה מימין) ששם למדדי PSNR ו SSIM יש ערכים גבוהים יותר אבל התמונה מטושטשת יותר.

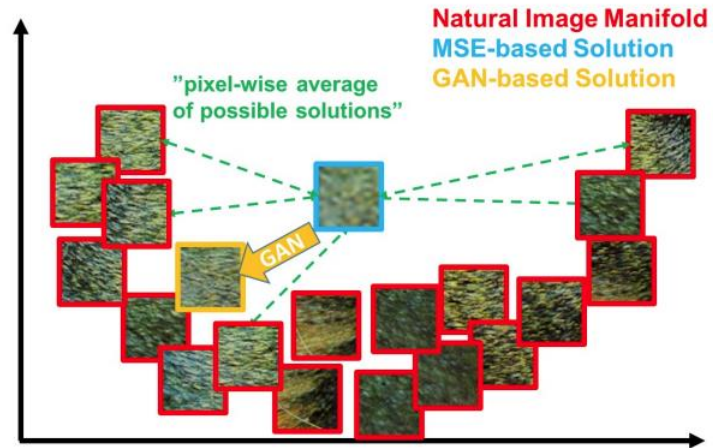


איור 15: המחשת שיפור המתקבל בשימוש ב Content loss. התמונה נלקחה מ[9].

4.1.1 דוגמה לשימוש בפונקציית הפסד לשמירה על תוכן התמונה

שימוש ברשת GAN וב content loss לצורך שיפור איכות תמונה ניתן לראות במאמר [10] אשר מציג את ארכיטקטורת SRGAN. המאמר מראה שיפור איכות תמונה של סופר רזולוציה, המטרה של שיפור זה היא להגדיל את רזולוציית התמונה. הגדלת הרזולוציה 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. לדוגמה הטקסטורה אשר אוחדה לכמה פיקסלים בודדים, אשר צריכה להתרחב למספר פיקסלים יותר גדול. כפי שהוסבר, שימוש בפונקציות הפסד אשר יינסו למזער את אפשרות השגיאה עבור פיקסל יקבלו תוצאות מטושטשות. הרעיון מאחורי העבודה ב [10] הוא לשחזר תמונה הנראית טוב לתפיסה האנושית, גם אם זה על חשבון של שינוי המידע בצורה מועטה.

באיור 16 מחברי המאמר מראים דגימה של הטקסטורות הנוצרות באימון GAN ואימון MSE. האיור מציג מקטעים שנלקחו מתמונה אמיתית (אדום), מקטע שנוצר על ידי אימון MSE (כחול), ומקטע שנוצר על ידי אימון רשתות יריבות (צהוב). ניתן לראות שאימון MSE יוצר תמונות שהם ממצעות את הפיקסלים ואילו אימון GAN מייצר מקטעים שדומים לטקסטורה של תמונת המקור.



איור 16: הדגמת הטקסטורות הנוצרות באימון GAN ואימון MSE. באיור רואים מגוון טקסטורות אפשריות עבור פאטץ מסויים הטקסטורות האפשריות מהעולם האמיתי מוצגות באדום, הטקסטורה הקרובה אליהם ביותר המתקבלת מאימון MSE מוצגת בכחול, והטקסטורה המתקבלת מאימון GAN מוצגת בצהוב. ניתן לראות שטקסטורה הנוצרת מאימון GAN קרובה יותר לטקסטורה אפשרית כלשהי מהעולם האמיתי. לעומתה טקסטורה הנוצרת באימון MSE לא קרובה לטקסטורה מהעולם האמיתי והיא שילוב של כמה טקסטורות אפשריות עברה, ולכן פחות חדה ויותר מטושטשת. התמונה נלקחה מ[10].

האלגוריתם דורש באימון תמונות באיכות רזולוציה גבוהה ותמונות באיכות רזולוציה נמוכה, על כך המחברים מציעים לייצר זוגות של תמונות. ניתן לייצר תמונה מוקטנת מתמונה מוגדלת בכמה דרכים כמו למשל על ידי החלקה של התמונה המוגדלת ודגימה כפי שהמחברים עשו במאמר.

במאמר משתמשים בשתי פונקציות הפסד אשר לחיבור ביניהם הם קוראים perceptual loss (מלשון תפיסה). החלק הראשון הוא שימוש ב content loss אשר הוצע במאמר [9] ומוסבר בסעיף 4.1 בעבודה זו. מטרת חלק זה היא לשמור על מבנה התמונה, כלומר לשמור על הפיצ'רים הבסיסיים בתמונת המקור.

החלק השני בפונקציית ההפסד הוא adversarial loss בחלק זה נעשה שימוש בערכי תוצאות ההסתברות של הרשת הדסקרימנטיבית 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 הדסקרימטור

$$L(G, D) = E_{I^{HR} \sim p_{train}}(\log(D(I^{HR}))) + E_{I^{LR} \sim p_G}(\log(1 - D(G(I^{LR}))))$$

I^{HR} - תמונות ברזולוציה גבוהה

I^{LR} - תמונות ברזולוציה נמוכה

פונקציית ההפסד הגנרטור באימון ה adversarial היא :

$$L_{generator}(I^{LR}, D, G) = - \sum_1^N \log (D(G(I^{LR})))$$

פונקציית ההפסד באימון הגנרטור :

$$L_{SR}(I^{HR}, I^{LR}, D, G) = L_{generator}(I^{LR}, D, G) + \lambda L_{content}(I^{HR}, I^{LR}, G)$$

כאשר במקרה זה $\lambda = 10^{-3}$

באיור 17 הלקוח ממאמר [10] רואים את ההבדל בין SRResNet ל SRGAN אשר שניהם תוצרים של אותה ארכיטקטורה של רשת ResNet [24] אשר יוסבר עליה בהרחבה בהמשך סעיף זה בעבודה. ארכיטקטורת SRGAN אומנה בתוספת של פונקציית הפסד adversarial loss . באימון SRGAN משתמשים ב Transfer learning לאתחול משקולות הגנרטור, תחילה מאמנים את הגנרטור ללא הדסקרימנטור בצורת SRResNet לאחר מכן ממשיכים את האימון בצורת הרשתות הירביות. ניתן לראות שאכן התמונה של SRGAN נוצרת חדה יותר. הסיבה לכך היא שהרשת הגנרטבית מייצרת תמונות אשר יקבלו הסתברות גבוהה ברשת הדסקרימנטבית אשר אומנה לסווג האם תמונה הגיעה מהעולם האמיתי או מהרשת הגנרטבית. שימוש בעקרון זה מייצר תמונות אשר נראות יותר דומות לתמונות מהעולם האמיתי, והדבר לפעמים מגיע על חשבון טעות בדיוק של שחזור הפרטים כפי שניתן לראות באיור. האיור מציג בנוסף את ערכי PSNR(db) וערכי SSIM על גבי כל תמונה שהגדילו לה את הרזולוציה, האיור מראה שערכי מדדים אלה לא בהכרח נותנת תוצאה גבוהה ביחס לחדות התמונה, נרחיב על כך בפרק 5 בעבודה זו.

ניתן לראות שהטקסטורה ב SRGAN אומנם חדה ונראית אמיתית, אבל לא נאמנה למקור, למשל באזור הבד שעל המצח והצוואר.



איור 17 : הדגמת איכות ועדיפות שימוש ב-SRGAN אשר בעל אותה ארכיטקטורה כמו SRResNet אבל אומן בעזרת אימון רשתות יריבות. התמונה נלקחה ממאמר [10].

רשת הגרנטור במאמר בנויה בצורת ארכיטקטורת ResNet [24] ו skip connections.

חיבור residual המוצג במאמר [24], הוא חיבור בין כניסת שכבה (או בלוק שכבות) למוצא שלה.

בעזרת חיבור שכזה נוצרת הקלה לרשת בייצור פונקציית הזהות במעבר בין השכבות. פונקציית הזהות היא פונקציה אשר אמור להיות קל לייצר אותה. עקרון זה מאפשר לשמור ערכים בסיסים לאורך הרשת, ורק להתאמן על ייצור שינויים תבניתיים.

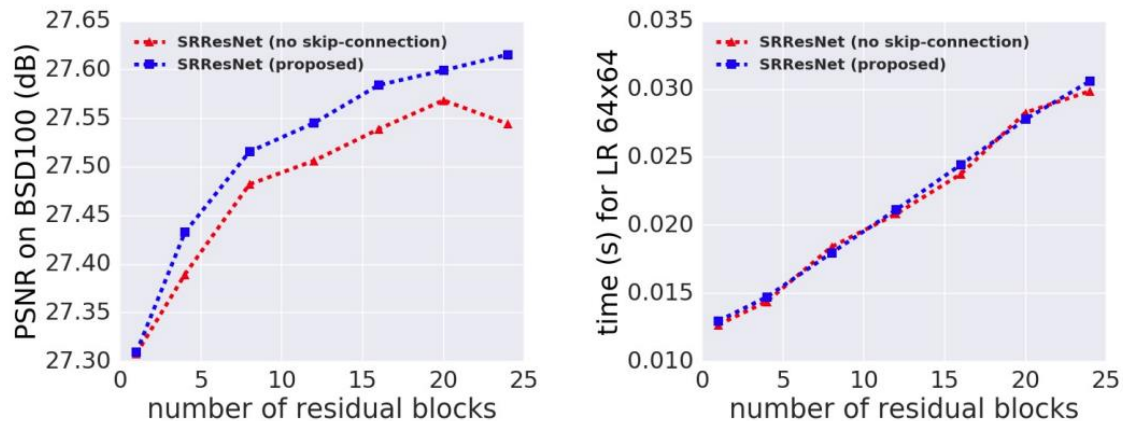
עקרון נוסף שחיבור residual ו skip connections עוזר הוא בהעלמות הגרדיאנט באלגוריתם gradient descent. באלגוריתם מנסים לחשב את כיוון שינוי המשקולות ברשת. שינוי המשקולות יהיה בכיוון השיפוע שיקטין את ערך פונקציית המטרה. חישוב הגרדיאנט בשכבות הראשונות מושפע מגרדיאנטים בשכבות הסופיות ולכן ככל שמוסיפים יותר 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.

בשל העקרונות הנ"ל ארכיטקטורת resnet מאפשרת להשתמש ביותר שכבות, שימוש ביותר שכבות מאפשר ליצור פונקציות מיפוי יותר מורכבות. בנוסף ככל שמוסיפים יותר שכבות כך זמן הפעלת הרשת גדל. שימוש בעקרון של skip connections יבוצע בכל הארכיטקטורות אשר אסקור בהמשך עבודה זו.

מחברי SRGAN השתמשו בארכיטקטורת resnet ו skip connections, וניסו למצוא את נקודת העבודה האופטימלית בין מספר השכבות אשר גורם להפעלה איטית של הרשת, למול שיפור באיכות התמונה. באיור 30 ניתן לראות את המדידות אשר בוצעו ככל שגדלו בכמות הבלוקים בחיבור

residual. ניתן לראות כי בהוספת שכבות בחיבור residual אכן יש השפעה חיובית על איכות התמונה שכן ערכי מדד PSNR עולים. (מדד PSNR מוסבר בסעיף 5.1 בעבודה זו).

מחברי המאמר הוסיפו חיבור נוסף בין הכניסה לבלוקי ה residual למוצאם. השפעתו מוצגת בגרף הכחול, הגרף האדום הוא שימוש בארכיטקטורה ללא חיבור זה. ניתן לראות כי חיבור זה לא משפיע על איכות התוצר שכן הגרף הכחול גבוה יותר בתמונה השמאלית. בנוסף ניתן לראות שאין לחיבור זה השפעה בזמן ריצה של הרשת שכן בגרף הימני לא רואים הבדל בין הגרפים



איור 30: משמאל, מדידה של כמות שכבות ה residual ביחס למדד איכות התמונה PSNR. מימין כמות שכבות ה residual ביחס לזמן הפעלת הרשת. התמונה נלקחה ממאמר [10]

CGAN - Conditional Generative Adversarial Nets 4.2

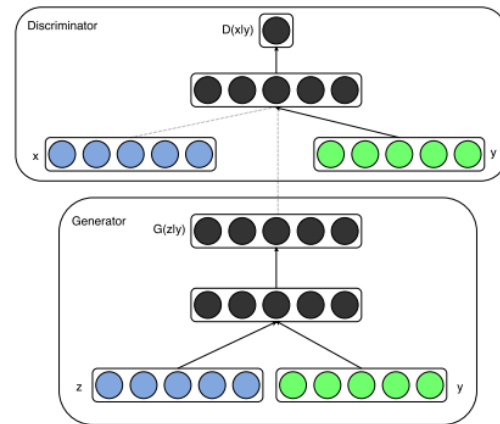
CGAN [12] מאפשר ליצור אובייקטים X שמגיעים ממחלקה Y ספציפית. העקרון נעשה על ידי הכנסת וקטור מידע נוסף המתקשר למחלקה אותה רוצים לייצר.

כעת הרשת הדסקרימינטיבית נותנת ערכים גבוהים רק עבור אובייקטים שנראים מהעולם האמיתי ושמתיימרים לווקטור המידע הנוסף כפי שנצפה בבסיס הנתונים. הרשת הדסקרימינטיבית לומדת להתאים בין אובייקטים מהעולם האמיתי לווקטור המידע הנוסף לכן על מנת שהרשת הגנרטיבית תצליח לגרום לעליה בפונקציית ההפסד, על הרשת הגנרטיבית להתאים את האובייקטים שהיא מייצרת לווקטור המידע הנוסף. פונקציית המטרה המעודכנת היא

$$L_{CGAN}(G, D) = E_{x, y \sim P_{data}(x, y)}(\log(D(x, y))) + E_{x \sim P_{data}(x), z \sim P_Z}(\log(1 - D(x, G(z, y))))$$

איור 18 מציג את מבנה הארכיטקטורות CGAN, ניתן לראות את וקטור התווית Y הצבוע בירוק מוכנס גם לגנרטור וגם לדיסקרימיננטור. בנוסף כמו בארכיטקטורות GAN קלאסיות, הרשת הגנרטיבית מקבלת וקטור רעש Z (מוצג בצבע כחול בריבוע התחתון) ומייצרת אובייקט. האובייקט הנוצר מהרשת הגנרטיבית (צבוע בשחור בריבוע התחתון) מוכנס לסירוגין עם אובייקט X מהעולם האמיתי (צבוע בכחול בריבוע העליון). וקטור של האובייקטים המתקבל לסירוגין מהעולם האמיתי ומהרשת הגנרטיבית

מצורף לווקטור התווית Y ומוכנס לרשת הדסקרימינטיבית (צבוע בשחור בריבוע העליון) שאמורה לקבוע האם אובייקט שנכנס אליה מתאים למחלקה Y והגיע מהעולם האמיתי.



איור 18: מבנה ארכיטקטורת CGAN, התמונה נלקחה מ[12].

4.2.1 דוגמא שימוש בארכיטקטורת CGAN לצורך שיפור איכות תמונה

שימוש ב CGAN לצורך שיפור תמונה ניתן לראות במאמר [6] המציג את ארכיטקטורת CANDY. המאמר מציע ארכיטקטורה להורדת רעש ערפל. תמונות אשר צולמו בתנאי מזג אוויר קשה כמו ערפל או אובך סובלות מהסתרות באזורים מסוימים, פחות חדות ובעלות צבעים לא ברורים.

מכיון שהערפל מסתיר את המידע האמיתי בחלקים מסוימים בתמונה, שימוש ברשתות אשר מנסות לשחזר את ערכי המקור באופן מדויק, יקבלו תוצאות מטושטשות, מאותן סיבות חוזרות אשר הסברנו בעבודה זו. מחברי המאמר מדגישים את העובדה שהם משתמשים ברשתות GAN על מנת ליצור תמונות אשר שמות דגש גם על אסטטיקת התמונה.

בארכיטקטורה ממאמר [6] הדיסקרימינטור מקבל בכניסה זוגות של תמונות כאשר לסירוגין הזוגות הן :

1. תמונות אמיתיות של סצנה עם ערפל וסצנה מתאימה לה ללא ערפל.
 2. תמונות אמיתיות של סצנה עם ערפל ותמונה שנוצרה מהרשת הגנרטיבית אשר קיבלה את התמונה המקורית עם הערפל ויצרה תמונה ללא ערפל.
- במאמר המחברים מתארים ששימוש בווקטור רעש כפי שמוגדר ב CGAN לא תרם, והרשת למדה להתעלם ממנו ולכן פונקציית ההפסד בה הם משתמשים לא מכילה ווקטור זה

$$L_{\text{CANDY}}(G, D) = E_{x, y \sim P_{\text{data}}(x, y)}(\log(D(x, y))) + E_{x \sim P_{\text{data}}(x)}(\log(1 - D(x, G(x))))$$

כאשר X מייצג תמונת מקור עם ערפל ו Y מייצג תמונת מקור ללא ערפל.

במאמר המחברים מוסיפים מספר פונקציות הפסד על מנת לגרום לאימון יציב יותר, הרעיון לשימוש בפונקציות האלה מוסבר בחלק קודם בעבודה זו:

- Content loss – עוזר למזער את המרחק של מאפיינים עיקריים של התמונה המיוצרת והמקורית
- L2 loss – עוזר למזער את המרחק בין הפיקסלים של תמונת המקור למוצא.
- Ls1 – העוזר בשמירה על מרחק בין פיקסלים, ובניגוד למרחק L2 פחות מעניש על מרחק גדול, דבר הגורם לפחות טשטוש.

פונקציית ההפסד המוצעת במקרה זה היא פונציקה המשתנה לפי ערך המרחק בין הפיקסלים של התוצר מרשת הגאן לתמונת התוצר הרצויה, אם המרחק לא גדול (נורמת L1 של המרחק קטנה מ 1), פונקציית ההפסד תהיה $0.5 d^2$. אחרת המרחק בין הפיקסלים גדול ואנחנו נרצה

להעניש בצורה שונה בעזרת הפונקציה $\|d\|_{L1} - 0.5$

$$d = Y - G(x)$$

$$L_{s1}(G) = (\|d\|_{L1} < 1) ? 0.5 d^2 : \|d\|_{L1} - 0.5$$

פונקציית ההפסד הסופית מוגדרת מחיבור של הפונקציות הפסד המתוארות בצורה הבאה

$$\argmin_G \argmax_D L(G, D) = L_{CANDY}(G, D) + L_{L2}(G) + L_{s1}(G) + L_{content}(G)$$



(b) Hazy input image

Ground-truth

CANDY

איור 19: תוצרי אלגוריתם CANDY, התמונה נלקחה מ[6]. ניתן לראות את תמונת הערפל, התמונה המקורית ללא הערפל, ותמונת המוצא של הארכיטקטורה.

4.3 Pix2Pix שינוי ערכי תמונה עבור מיפוי מוצג

הארכיטקטורה Pix2Pix אשר מתוארת במאמר [7], מאפשרת לייצר מיפוי בין תמונות לפי שינוי כללי כרצוננו. זאת בשונה מהארכיטקטורות GAN אשר מקבלות ווקטור רעש ומטרותן לייצר אובייקט מהתפלגות מסוימת. ארכיטקטורה זו מקבלת סקיצה של תמונה או בסיס תמונה קיימת עליה רוצים לבצע שינוי, על בסיס התמונה הקיימת הארכיטקטורה יוצרת תמונות קרובות לתמונת המקור מבחינת בסיס התמונה ובתוספת שינויים ככל שנרצה. המחברים הדגימו כי הארכיטקטורה שלהם מהווה שיטה

כללית לשינוי ערכי תמונה. המחברים מדגימים במאמר מגוון שימושים של מיפוי תמונה לתמונה בתוצאות טובות למול ארכיטקטורות מתחרות אשר נתפרו במיוחד למשימות הספציפיות אותן הן באות לפתור. בארכיטקטורה Pix2Pix מתקבלות תוצאות טובות עבור מגוון משימות ללא צורך בשינוי מבנה הארכיטקטורה או שינוי פונקציית ההפסד.

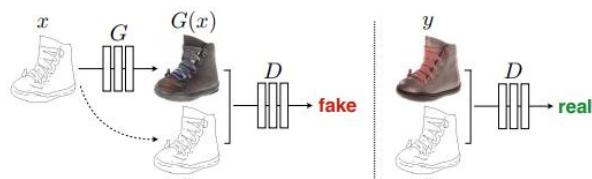
להלן שיפורי איכות התמונה אשר הודגמו במאמר [7]:

- מיפוי תמונות חום חסרות צבע לתמונת מקור.
- תמונות אפורות שמומרות לתמונות צבע.
- השלמת פיקסלים חסרים בתמונה.

עבור ארכיטקטורות GAN אשר סקרנו עד כה בעבודה, היתה דרישה בקבלת ווקטור רעש אשר ייצר תוצאות שונות עבור כניסות שונות. מחברי המאמר טוענים שבכל ניסוי אשר הם עשו, הם לא ראו השפעה של ווקטור הרעש על תוצאות הארכיטקטורה שלהם. במקום להכניס ווקטור רעש, מחברי המאמר הצליחו לייצר תוצאות שונות עבור כניסות דומות בעזרת הוספת שכבות dropout. הוספת שכבות dropout היא שיטת רגולציה אשר בה בשלב האימון עבור כל שכבה מחליטים אם לבטל את השפעת חלק מהנוירונים בשכבה בהסתברות ברנולי הנקבעת כהיפר פרמטר עבור כל שכבה בנפרד. לרוב בשימוש בשיטת רגולציה זו, לאחר שלב האימון, מבטלים את השימוש בשכבה ומפעילים את הארכיטקטורה בלעדייה. במאמר המחברים משאירים את השפעת שכבת ה dropout גם לאחר שלב האימון, וזאת לצורך ייצור תמונות עם מאפיינים סטוכסטיים שונים עבור כניסות קבועות.

מטרת הארכיטקטורת Pix2Pix היא לייצר שינוי על בסיס תמונה קיימת, בעזרת מדידות, מחברי המאמר מוכיחים כי שימוש בארכיטקטורת CGAN עדיפה על פני שימוש בארכיטקטורת GAN לצורך שמירה על מבנה תמונת המקור. מחברי המאמר מציינים שהסיבה לכך היא שארכיטקטורת GAN שמה דגש על איכות התמונה בלבד ואילו ההתניה על תמונת המקור (שהוא ווקטור המיע הנוסף) בשיטת CGAN מאפשר לרשת להתמקד בייצור תמונה הדומה לתמונת המקור ורק ללמוד את המיפוי הרצוי. המחברים מציינים ששימוש במבנה GAN גרם להם לייצר תמונות הנראות טוב אבל מגיע יותר לקריסות מצב. ארכיטקטורת Pix2Pix דומה לארכיטקטורת CGAN שתוארה בסעיף 4.2 בעבודה זו, באופן בו הארכיטקטורה מקבלת ווקטור תנאי בכניסה לרשת הגנרטיבית ולרשת הדסקרימנטיבית. מטרת הרשת הגנרטיבית היא לייצר אובייקטים מתאימים לווקטור התנאי, מטרת הרשת הדסקרימנטיבית היא לסווג האם תמונה המתקבלת בכניסה ווקטור התנאי תואמים. ההבדל בכניסות בין הארכיטקטורות הוא שבארכיטקטורת Pix2Pix ווקטור התנאי הוא יותר אינפורמטיבי והוא מתבסס על תמונות בגדול מלא אותן רוצים לשפר, הבדל נוסף הוא שארכיטקטורה זו לא מקבלת ווקטור רעש בכניסה, הסיבה לכך הוסברה בפסקה למעלה.

באיור 20 ניתן לראות דוגמא לכניסות לרשת הגנרטיבית והדסקרימניבית בהתאמה בארכיטקטורת Pix2Pix בחלק השמאלי ניתן לראות את הרשת הדסקרימניבית D מקבלת את ווקטור התנאי האינפורמטיבי של קווי המתאר ואת תוצרי הרשת הגנרטיבית שנוצרו אחרי הפעלת הרשת הגנרטיבית על אותו ווקטור תנאי של קווי מתאר. בחלק הימני בתמונה ניתן לראות את פעולת הרשת הדסקרימניבית על כניסה של תמונה אמיתית ווקטור תנאי תואם לה.



איור 20: מבנה רשת Pix2Pix, התמונה נלקחה מ [7]. גם הגנרטור וגם הדיסקרימנטור מקבלים את ווקטור התמונה של קווי המתאר. נשים לב כי בניגוד לאיור מארכיטקטורת CGAN, במקרה זה לא מצורף ווקטור רעש.

מחברי Pix2Pix משתמשים במדידת מרחק הפיקסלים בין תמונת אליה רוצים להגיע לתמונת המוצא של הגנרטור, מכיוון שאנחנו רוצים ליצור תמונה יחסית דומה לתמונה אליה רוצים להגיע בתוספת שינויים קטנים, אנחנו יכולים למדוד כמה המוצא של הרשת הגנרטיבית דומה לתמונה אליה רוצים להגיע, את ערך ההפרש מוסיפים כאיבר רגולציה בפונקציית ההפסד של רשת הגנרטור.

$$L_{L1}(G) = E_{x,y \sim P_{data}(x,y)} (||Y - G(x)||_1)$$

הערכים האופטימליים של פונקציית המטרה עבור הגנרטור:

$$\argmin_G \argmax_D L(G, D) = L_{cGan}(G, D) + L_{L1}(G)$$

כאשר החלק השמאלי הוא פונקציית המטרה של CGAN והחלק הימני הוא תנאי רגולציה על המרחק בין הפיקסלים מהרשת הגנרטיבית לאובייקט שרוצים ליצור מהעולם האמיתי. נשים לב שלא השתמשנו פה בווקטור הרעש כפי שמופיע בפונקציה ההפסד ב CGAN, מכיוון שמחברי המאמר מתארים שהוא לא תרם אינפורמציה לארכיטקטורה.

פונקציית ההפסד עבור הדיסקרימנטור נשארת כפונקציית ההפסד המקורית של רשת הדיסקרימנטור בארכיטקטורת CGAN, הפונקציה $L_{cGan}(G, D)$.

כפי שהסברנו בסעיף 4.1, שימוש בפונקציית מרחק פיקסלים מייצר תמונה מטושטשת. מחברי המאמר מודעים לעובדה זו ומציינים כי השימוש בתנאי רגולציה כזה אכן שומר על תדרים נמוכים יותר בתמונה וגורם לתדרים גבוהים להיאבד, כלומר ליצור תמונה מטושטשת. כפי שהמחברים מתארים תנאי

רגולציה זה נוסף על מנת לגרום לגנרטור להוציא תמונות מוצא דומות לתמונה אליה רוצים להגיע. על מנת לשמור על התדרים הגבוהים מחברי המאמר משתמשים בארכיטקטורת דיסקרימיננטור חדשה אשר מטרתה לסווג האם מקטעים של התמונה נראים אמיתיים. לארכיטקטורת דיסקרימיננטור זו קוראים PatchGan. דסקרימיננטור זה ראשית מסווג עבור כל מקטע בתמונה האם הגיע מהעולם האמיתי או מהרשת הגנרטיבית. לאחר סיווג המקטעים, מבצעים מיוצע על התוצאות לקבלת ערך סיווג הדסקרימיננטור על התמונה. צורה זו של הדיסקרימיננטור אשר מחזיר מספר ערכים עבור אזורים שונים מאפשרת לרשת ה GAN ללמוד הרבה יותר מידע מאשר דסקרימיננטור רגיל שמחזיר תשובה כללית אחת לגבי התמונה. בצורה זו ניתנת לרשת היכולת לבחון אזורים ספציפיים כמה הם אמיתיים או לא.

מכיוון שהארכיטקטורה של הדיסקרימיננטור בנויה משכבות קונבולוציה, שימוש בדסקרימיננטור שכזה יוצר יתרון לארכיטקטורה בכך שעבור כל גודל של תוצר רשת גנרטור נוכל לקבוע גודל וארכיטקטורה קבועים לדיסקרימיננטור, היתרון הוא בכך שעבור תמונות מוצא גדולות נוכל להשתמש בארכיטקטורת דיסקרימיננטור קטנה יותר בעלת פחות משקולות, לכן האימון יהיה יעיל יותר.

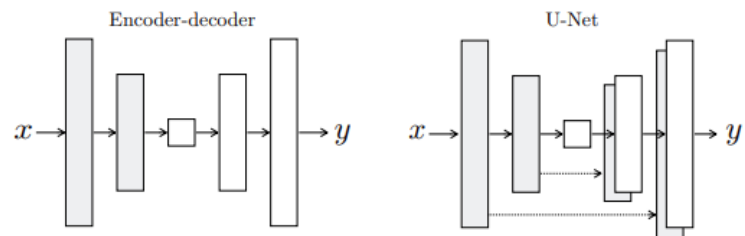
באיור 21 ניתן לראות דוגמא עבור כמה תוצרי אימון עבור דיסקרימיננטורים שאומנו עם גדלי מקטעים שונים. בתמונה השמאלית ניתן לראות אימון של רשת הגנרטור רק בעזרת פונקציית הפסד $L_{L1}(G)$. בתמונה השנייה משמאל ניתן לראות אימון דסקרימיננטור עם מקטע בגודל 1×1 , ניתן לראות כי הדבר מוסיף לצביעת התמונה. שימוש במקטע בודד יוצר תמונה מטושטשת למדי. התמונה המרכזית אומנה עם מקטעים בגודל 16×16 , ניתן לראות כי התמונה מתחדדת יותר מהתמונה שאומנה עם מקטע בגודל 1×1 , בתמונה זו ניתן לראות שעדיין נוצרים תוצרים עם שינויים לא רצויים. בתמונה השנייה מימין ניתן לראות אימון של מקטעים בגדלי 70×70 פיקסלים, זהו הערך אשר מחברי המאמר השתמשו בו לשאר הניסויים להוכחת יכולות הארכיטקטורה. ניתן לראות כי תוצרי הרשת ברורים וחדים. בתמונה הימנית ביותר ניתן לראות אימון של מקטעים בגודל 286×286 . לפי הניסויים שהמחברים ביצעו, אימון של דסקרימיננטור בעל מקטעים בגודל 286×286 פיקסלים, יוצר תוצאות טובות פחות, המחברים טוענים שיש אפשרות שהירידה בתוצאות התוצרים מתרחשת מכיוון שלרשת דסקרימיננטור זו יש הרבה משקולות לטייב ולכן ההתכנסות שלה לתוצאות טובות הייתה יותר קשה.

לדוגמה המוצגת באיור 21 נמדדה מטריקת FCN אשר מתוארת בטבלה [2] בסעיף 5.3.2 בעבודה זו. המטריקה תואמת לתוצאות המתקבלות, הערכים עבור הדיסקרימיננטור עם מקטעים בגדלי 70×70 מקבלים תוצאות טובות ביותר.



איור 21: הדגמת תוצרי ארכיטקטורת Pix2Pix עבור גדלי מקטעים שונים בדסקרימנטור PatchGan. התמונה נלקחה מ [7].

מחברי Pix2Pix מבצעים שימוש בגנרטור אשר בנוי בצורה של רשת UNET [20]. במקור ארכיטקטורת רשת UNET נועד לסגמנטציה, כקלט הוא מקבל תמונה וכפלט הוא מוציא סגמנטציה של תמונה זו, מחברי Pix2Pix משתמשים בארכיטקטורה זו מכיוון שהיא טובה בליצור מיפוי בין תמונות. הארכיטקטורה בנויה באופן בו המידע הנכנס לרשת עובר צמצום בעזרת הורדת גדלי השכבות ולאחר מכן המידע נפרש מחדש על ידי הגדלות שכבות. בצורה זו (הדומה בבסיסה ל AUTO ENCODER) נוצר מעין קידוד של המידע בשכבות הפנימיות וכאשר המידע נפרש מחדש רק חלק ממנו נשמר. על מנת לשמור על יותר פרטים בזמן הדחיסה והפרישה, משתמשים בחיבורים בין השכבות המאפשרים למידע לדלג אל המשך השכבות הפנימיות. עקרון נוסף שהחיבורים בין הרשתות עוזר הוא בשלב האימון, בהתגברות על העלמות הגרדיאנט. רשתות עם שכבות רבות נוטות להיכנס למצב של העלמות גרדיאנט בשלב האימון, כאשר פעפוע לאחור של המידע על הגרדיאנט מהשכבות הסופיות צריך לעבור הרבה שכבות הוא נהיה פחות רלוונטי עד שהוא מגיע לשכבות הראשונות. החיבורים הנוספים ב UNET עוזרים במעבר מידע הגרדיאנט לשכבות הראשונות מהר יותר. בסעיף 5.3.2 נציג את המטריקות ודוגמאות אליהן, אשר בעזרתן מחברי Pix2Pix החליטו לבחור את ארכיטקטורת הגנרטור כ UNET ואת פונקציית ההפסד המתאימה לארכיטקטורה זו.



איור 22: התמונה נלקחה מ [7] והיא מראה את ארכיטקטורת UNET ו ENCODER DECODER. במאמר הראו כי החיבורים בין השכבות בארכיטקטורת UNET עזרו לשמר מידע בעזרת הקישור בין השכבות, שבמקרה שמשתמשים רק ב ENCODER DECODER נעלם.

4.3.1 דוגמא לשימוש patchGan

DeblurGAN [8] מראה שימוש בכמה עקרונות מהארכיטקטורות האחרונות שדיברנו עליהם לצורך הורדת טשטוש מתמונה. הטשטוש אשר אותו משפרים הוא כזה שיכול להיווצר מצילום בזמן תזוזת מצלמה. המאמר עושה שילוב בין עקרונות מכמה מאמרים אשר סקרנו בסעיפים קודמים.

עבור אימון הגנרטור המחברים משתמשים בארכיטקטורת resnet בשילוב perceptual loss כפי שהוצג במאמר SRGAN. עבור הדסקרימנטור המחברים מבצעים שימוש בארכיטקטורת PatchGan וב CGAN בדיוק כפי שמוצע במאמר Pix2Pix אשר תואר קודם בעבודה זו.

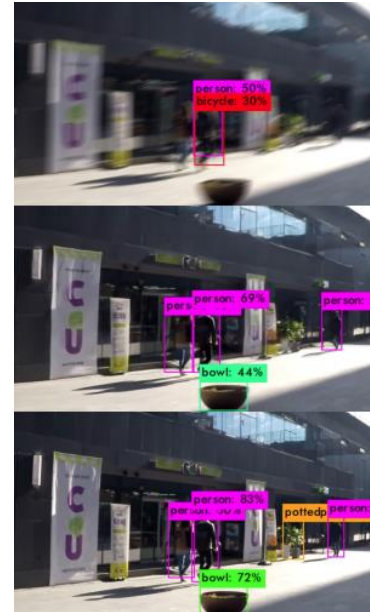
בנוסף מאמר זה מבצע שימוש ב-WGAN-GP אשר על פי מחברי המאמר הוריד את כמות קריסות המצב ונתן את האפשרות לאמן ארכיטקטורה קטנה בהרבה מהארכיטקטורה הקודמת אשר שימשה להורדת טשטוש. הקריטיק במקרה זה מקבל את התמונה החדה המקורית והתמונה המטושטשת ומחשב את מרחק EM ביניהם.

מחברי המאמר מתארים שני ניסויים אשר נעשו בבחירת הארכיטקטורה. הניסוי הראשון הוא להחליף את עקרון השימוש ב-perceptual loss בפונקציית ההפסד לשימוש ב-MSE, אשר גרם לתוצרי הגנרטור לא להתכנס לתמונות בעלות משמעות. הניסוי השני היה להוריד את השפעת החלק בפונקציית ההפסד אשר מגיע מהאימון של הרשתות היריבות, במקרה זה תוצרי הרשת היו בעלי משמעות אבל נראו מטושטשים. שני הניסויים מאוד תואמים את העקרונות עליהם דיברנו בעבודה זו. העקרון הראשון שימוש ב-MSE לצורך מזעור שגיאת השחזור יוצר תמונות אשר הן מיצוע של האפשרויות, וייתכן כי מיצוע זה לא מהווה אובייקט מהעולם האמיתי. העקרון השני, שימוש באימון רשתות יריבות, דוחף את התוצרים של הגנרטור לייצר טקסטורות חדות.

האלגוריתם דורש אימון בעזרת זוגות של תמונות מטושטשות וחדות מאותה סצנה. אין דרך קלה ליצור זוגות שכאלה, על מנת ליצור זוגות לצורך האימון המחברים משתמשים בכמה שיטות ליצור זוגות שכאלה. חלק מהשיטות כוללות מצלמה בחשיפה מהירה בתזוזה וחיבור אינטרפולציה של התמונות, וחלק מהשיטות הן הפעלת קונבולוציה של קרנל תזוזה על תמונת המקור.

לצורך הערכת טיב שיפור הורדת הטשטוש, מחברי המאמר הציעו עקרון של מדידת טיב איכות התמונה על ידי שימוש ברשת דיטקציה שאומנה לזהות אובייקטים בתמונה. המדדים המשווים במטריקה שלהם הם precision אשר יהיה גבוה ככל שזיהוי העצמים על ידי רשת הדיטקציה יהיה תואם לעצמים בתיוג המקור, recall אשר יהיה גבוה יותר ככל שנזהה יותר עצמים, $F1$ שהוא הממוצע ההרמוני של שני המדדים precision ו-recall. המחברים מראים כי האלגוריתם שלהם מקבל את התוצאות הטובות ביותר על סטטיסטיקות $F1$, recall, ביחס למתחרים, אבל בprecision התוצאות במעט פחות טובות מהמתחרים. ההסבר של המחברים ל precision הנמוך במעט הוא מכיוון שבאלגוריתם שלהם התמונה חדה יותר, לכן יש אפשרות שרשת הדיטקציה תזהה יותר עצמים קטנים, שיכול להיות וחלקם יהיו לא נכונים. לעומת זאת ניתן לראות כי במדדי ה-recall ו- $F1$ אלגוריתם שלהם מקבל את התוצאה הטובה ביותר מבין המתחרים, ולכן האלגוריתם שלהם עדיף.

באיור 23 ניתן לראות את שיפור איכות זיהוי העצמים אשר עולה בהפעלת אלגוריתם DeblurGAN על תמונה מטושטשת. התמונה התחתונה היא תמונת המקור, התמונה האמצעית התמונה לאחר השחזור, והתמונה העליונה היא התמונה שעברה טשטוש עליה הופעל האלגוריתם.



איור 23: הדגמת תוצאות אלגוריתם DeblurGAN, התמונה נלקחה מ[8].

4.4 CycleGAN שינוי ערכי תמונה עבור מיפוי אשר נלמד על ידי הארכיטקטורה

הרעיון שמסתתר מאחורי Pix2Pix הוא שבשלב האימון צריך לספק ווקטור תנאי אינפורמטיבי אשר ייתקבל בכניסה לארכיטקטורה ויהיה מתואם לתוצר הסופי אותו רוצים לייצר. על מנת ליצור ווקטור כניסה מתואם יש צורך במיפוי ספציפי בין ווקטור התנאי לתוצר הסופי הרצוי. מיפויים לדוגמא שהודגמו ב-Pix2Pix היו סגמנטציה מול תמונה אמיתית, או קווי מתאר שתואמים לתמונה הסופית שרוצים לייצר. לא תמיד ניתן למצוא מיפוי בין תמונת מקור לווקטור התנאי, למשל עבור בעיה של העברת סגנון מצייר אחד לצייר שני, לרוב אין דרך להתאים ווקטור תנאי פשוט שיתאר את המעבר בין סגנון הציירים, לכן לא נוכל ליצור התאמה של ווקטור תנאי מצייר אחד למוצא רצוי מצייר אחר.

במאמר [13] פיתחו שיטה של מיפוי בין מקבץ תמונות בסגנון אחד למקבץ תמונות בסגנון אחר ללא צורך בהתאמה של תמונות בין המקבצים בזמן האימון. על מנת לעמוד בדרישה שכל אובייקט ימופה לאובייקט שמתאים לו במקבץ השני הם משתמשים בשתי רשתות GAN אשר עובדות בצורה הפוכה, האחת מייצרת תמונות ממקבץ א למקבץ ב והשנייה מייצרת תמונות ממקבץ ב למקבץ א.

$$L_{\text{Gan}}(G, D_y) = E_{x \sim P_{r1}} E(\log(D_y(x))) + E_{z \sim P_z} (\log(1 - D_y(G(z))))$$

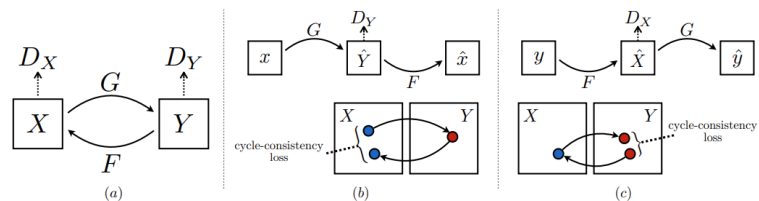
$$L_{\text{Gan}}(F, D_x) = E_{y \sim P_{r2}} E(\log(D_x(y))) + E_{z \sim P_z} (\log(1 - D_x(G(z))))$$

באופן תאורטי, בעזרת רשתות GAN ניתן לייצר מיפוי ממרחב P_{r1} למרחב P_{r2} , אולם ניתן לצפות כי רשת בעלת מספר רב של פרמטרים תוכל ללמוד מגוון מיפויים שונים, במקרה שלנו נרצה לגרום שעבור אובייקט יחיד x_i נוכל לכפות שיתאים לאובייקט y_i ספציפי. כלומר נרצה שלמיפוי תהיה משמעות, את

שמירת המשמעות עבור המיפוי מקבלים בעזרת עקרון המעגליות. עקרון זה מוסיף למנוע קריסת מצב, שכן מהדרישה מתקבל שלא כל האובייקטים ממקבץ אחד ימופו למקבץ אחר.

על מנת לגרום למיפוי F ומיפוי G להתאים אחד לשני, מוצע תנאי נוסף לפונקציית ההפסד אשר נעזר בעקרון של המעגליות בתרגום. בעקרון המעגליות Cycle Consistency כפי שמוצג באיור 24, אובייקט העובר ברשת הראשונה ממקבץ A למקבץ B ולאחר מכן מועבר שוב ברשת השנייה ממקבץ B למקבץ A , אמור להראות כמו אובייקט המקור. עקרון זה מוכנס כתנאי לפונקציית ההפסד בצורה הבאה:

$$L_{cyc}(G, F) = E_{x \sim P_{r1}}(\|F(G(x)) - x\|_1) + E_{y \sim P_{r2}}(\|F(G(y)) - y\|_1)$$



איור 24 : הצגת ארכיטקטורת Cycle Gan ועקרון ה Cycle Consistency. התמונה נלקחה מ[13]. התמונה מתארת את המיפוי מ 2 מקבצים X ו Y הרשתות F, G הגנרטורים D_x ו D_y הדיסקרימינטורים. (a) מראה את מיפוי X למרחב Y על ידי הגנרטור G , ואת דיסקרימינטור D_y שמקבל את התמונה ממרחב Y האמיתי ומהתמונות ש G מייצר. באותו אופן עם D_x, G, Y . (b) תיאור Cycle Consistency על מרחב X על ידי מעבר דרך מרחב Y וחזרה ל X . (c) תיאור Cycle Consistency על מרחב Y על ידי מעבר דרך מרחב X וחזרה ל Y .

רעיון נוסף שמחברי המאמר מציעים הוא שימוש במיפוי זהות Identity mapping והעקרון הוא שהפעלת רשת על תמונה ממקבץ מסוים, כאשר הרשת אמורה לבצע פעולה של המרה למקבץ המסוים ממנו התמונה הגיע, אמור לקבל תוצאה של תמונה זהה לתמונת המקור שנכנסה לרשת. גם כאן מכניסים תנאי זה לפונקציית ההפסד כאשר משתמשים במרחק פיקסלים לבדיקה של עד כמה התמונה משתנה במיפוי, המחברים במאמר מראים כי עקרון זה מוסיף יציבות למיפוי בין צבעים בין המקור למוצא.

$$L_{identity}(G, F) = E_{x \sim P_{r1}}(\|F(x) - x\|_1) + E_{y \sim P_{r2}}(\|G(y) - y\|_1)$$

על מנת לבצע את כל האמור לעיל יש לנו צורך בארכיטקטורה אשר מבצעת את כל העקרונות המדוברים, על כן משתמשים בפונקציית הפסד המשלבת מספר תנאים, 2 פונקציות ההפסד של רשתות הGAN אשר ממפות אובייקטים ממרחב X ל Y וממרחב Y ל X , בנוסף תנאי הרגולציה שמשמרים את Identity mapping לכל מרחב, ותנאי הרגולציה שמשמרים את עקרון ה Cycle Consistency במעבר בין המרחבים וחזרה.

הערכים האופטימליים של פונקציית המטרה:

$$\begin{aligned} \argmin_{G,F} \argmax_{D_x,D_y} L_{\text{CycleGAN}}(G, F, D_x, D_y) \\ = L_{\text{Gan}}(G, D_y) + L_{\text{Gan}}(F, D_x) + \lambda L_{\text{cyc}}(G, F) + \delta L_{\text{identity}}(G, F) \end{aligned}$$

במאמר נעשה שימוש בעקרון נוסף על מנת להגיע לייצוב האימון והוא לעדכן את הדסקרימינטור באמצעות שילוב של היסטוריה של תוצרי הגנרטור ותמונות מהעולם האמיתי. כפי שתיארנו במקטע 2.3 בעבודה זו, חלק מן הבעיות באימון נובעות מזה שאימון הרשתות היריבות נעשה באופן איטרטיבי לסירוגין, פעם על הרשת הדסקרימינטיבית ופעם על הרשת הגנרטיבית. בכל איטרציה אנו מעדכנים את משקולות הרשת הדסקרימינטיבית לפי מקבץ דוגמאות מהעולם האמיתי ומקבץ דוגמאות מהרשת הגנרטיבית במצב הנוכחי שלה. במצב שכזה הרשת הדסקרימינטיבית יכולה לעדכן את המשקולות כך שעבור דוגמאות שנלמדו כטעות בעבר יכולות להיחשב כאמיתיות בשלב הנוכחי, כלומר הרשת יכולה "לשכוח" את דוגמאות הטעות מהעבר. במאמר המחברים משתמשים בהיסטוריה של חמישים תמונות שנוצרו על ידי הגנרטור, ועל פי עקרון עדכון דגימה התמונות מוחלפות לסירוגין, כל פעם נשארות חצי מהתמונות ההיסטוריות ומוחלפות בחצי מהתמונות מתוצרי הגנרטור במצב הנוכחי.

4.4.1 דוגמה לשימוש בארכיטקטורת CycleGAN ובperceptual loss

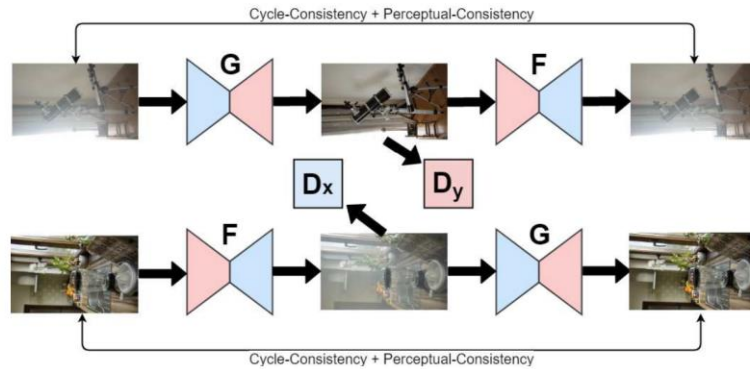
בארכיטקטורת Cycle-Dehaze [14] ניתן לראות דוגמה לשיפור איכות תמונה המשתמש בעקרונות מארכיטקטורת CycleGAN. המאמר מצליח להוריד ערפל מתמונה ללא צורך בין התאמת זוגות של תמונות עם ערפל לתמונות מאותה סצנה ללא ערפל, התאמה זו קשה להשגה. המאמר מוסיף שימוש בעקרון perceptual consistency המחובר בין עקרון Cycle Consistency, ל content loss. הסיבה לשימוש בעקרון perceptual consistency היא מפני שתמונת ערפל מאבדת מידע מהתמונה המקורית ולכן שימוש נורמה L1 להשוואה עם הפיקסלים מתמונת המקור, כפי שנעשה ב Cycle Consistency לא מספיק. עכשיו לשמירה על עקרון המעגליות משתמשים בעקרון דומה ל content loss, משתמשים בערכי השכבות הפנימיות ברשת סיווג כגון VGG אשר הופעלה על התמונה ממנה רוצים להוציא ווקטור של המאפיינים. להשוואת ווקטור המאפיינים בעקרון המעגליות משווים את מאפייני התמונה המקורית ומאפייני התמונה אשר הופעלו עליה שתי מיפויים של רשתות הגנרטורים, אשר ממפים למרחב האחר וחזרה למרחב המקורי.

עבור שכבה i המרחקים בין ווקטורי הפיצ'רים מחושבים

$$\begin{aligned} L_{\text{Perceptual}}(G, F) \\ = E_{x \sim P_{r1}} (||\phi_i(F(G(x))) - \phi_i(x)||_2) + E_{y \sim P_{r2}} (||\phi_i(F(G(y))) - \phi_i(y)||_2) \end{aligned}$$

פונקציית המטרה הכללית של הארכיטקטורה :

$$L_{\text{Cycle-Dehaze}}(G, F, D_x, D_y) = L_{\text{CycleGAN}}(G, F, D_x, D_y) + \lambda L_{\text{Perceptual}}(G, F)$$



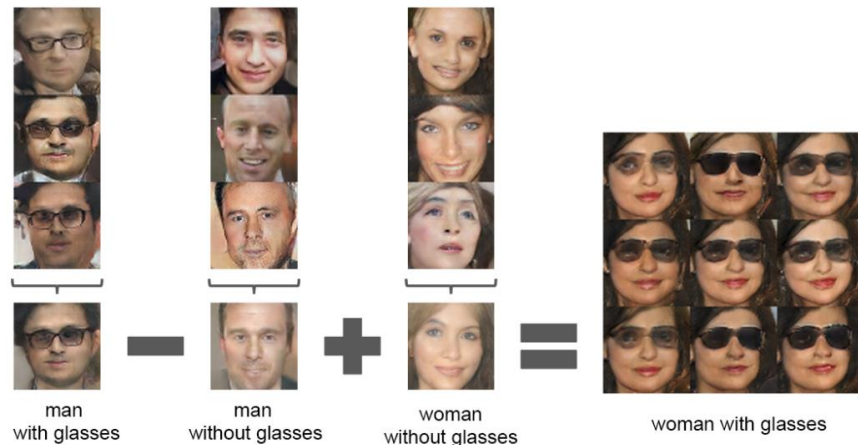
איור 25: הצגת ארכיטקטורת Cycle-Dehaze, המשתמשת ב perceptual consistency. האיור נלקח ממאמר [14]. תמונה ממרחב ללא ערפל ממופה למרחב עם ערפל וחזרה והמרחק מהתמונה המקורית והתמונה המומרת נמדד ב cycle consistency, perceptual consistency.

4.5 ייצור שינויים נשלטים בתמונה על ידי שימוש בשכבה הלטנטית.

מחברי מאמר [23] אשר תואר בסעיף 3.1 בעבודה זו, הראו לראשונה כי ניתן להסתכל על המיפוי שהארכיטקטורה לומדת ממרחב ווקטור הרעש Z לתוצרי הרשת הגנרטיבית כמרחב לטנטי רציף. המחברים מראים כי מעבר של אינטרפולציה לינארית בין נקודות במרחב הרעש מייצר תוצרים במרחב הפלט של הרשת הגנרטיבית, אשר השינוי עליהם משתנה באופן רציף ככל שמתקדמים באינטרפולציה. במאמר [15] הראו כי שינוי וקטור הרעש לכיוון מסוים במרחב יוצר שינוי בפיצ'ר מסוים. המטרה היא שהמיפולציה על שינוי וקטור Z תשפיע כך שפיצ'רים לאובייקטים הנוצרים יהיו בעלי תכונות שאנו נרצה לשנות.

מאמר [15] מציע דרך לבצע שינויים על וקטור Z בעזרת מניפולציות וקטוריות, הדרך שמוצעת שם למציאת כיוון וקטורי לתכונה היא לקחת את הווקטור הממוצע של כל האובייקטים בעלי אותה תכונה. במאמר הם מראים שניתן לבצע פעולות אלגבריות של חיבור וחסור לכיוונים של תכונות בעזרת הווקטורים הממוצעים שיוצא מהתכונות.

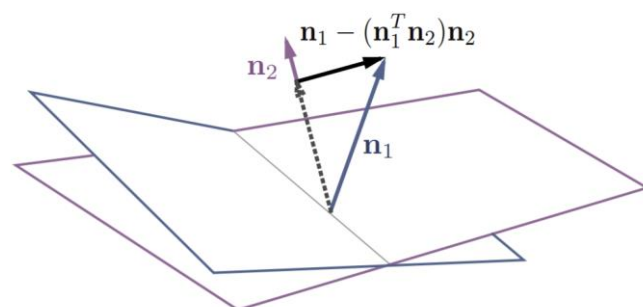
לדוגמה באיור 26, ניתן לראות שילוב של פעולות אלגבריות על השכבה הלטנטית על מנת לייצר תמונה של אישה עם משקפיים. ייצור תמונה של אישה עם משקפיים יכול להיווצר על ידי לקיחת ערך ממוצע מווקטורי רעש המייצרים תמונת גבר עם משקפיים, החסרת הווקטור של ממוצע של תמונות של גברים והוספת וקטור ממוצע של תמונות נשים.



ב

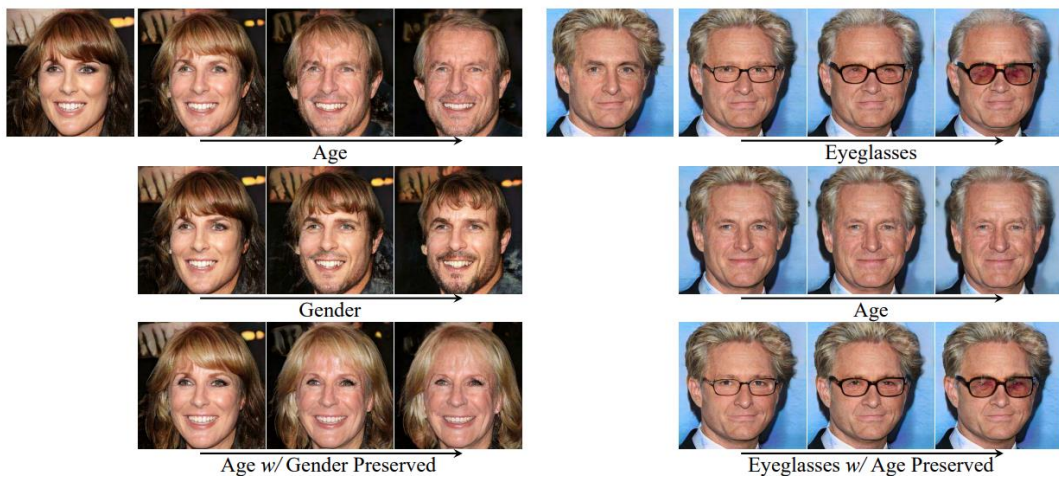
איור 26: שימוש בפעולות אלגבריות על ווקטורים המייצגים איברים בעלי תכונות רצויות במרחב הלטנטי. האיור נלקח מ [15]

האתגרים בעקרון Controllable generation הם שלפעמים כאשר רוצים לשנות פיצ'ר אחד הדבר יגרום לשינוי בפיצ'ר נוסף אשר הרשת הגנרטיבית איחדה בהקשר (entanglement) מכיוון שהפיצ'רים עצמם קשורים לרוב בעולם האמיתי, ולכן הרשת עצמה למדה את ההקשר של הפיצ'רים בשלב האימון (לדוגמא זקן לרוב מתקשר עם פנים גבריות). סיבה נוספת לקישור בין פיצ'רים יכולה להיווצר כאשר אנו מנסים למפות מרחב לטנטי קטן מידי יחסית לכמות הפיצ'רים הכללית במרחב. במקרה זה לרשת אין אפשרות לא לקשר בין פיצ'רים והמשקולות הנלמדים של פיצ'רים מסוימים יהיו תלויים אחד בשני. [16] מציע דרך להתקדם בכיוון פיצ'ר ספציפי שבחרנו ולבטל התקדמות בפיצ'רים שלא רצינו, בעזרת הבנה של המרחב הלטנטי על ידי חלוקה לכמה היפר מישורים אשר כל אחד מהם מפריד בין תכונה עם קלסיפיקציה בינארית. את המישורים ניתן למצוא למשל בעזרת SVM לכל פיצ'ר על המרחב הלטנטי כפי שנעשה במאמר [16]. במידה שנרצה לשנות תכונה של התמונה אשר משפיעה על תכונות נוספות, נוכל להתקדם רק במישור התכונה הרצויה על ידי מציאת וקטור אורטוגונלי לווקטור הנורמלי של המישור התכונה אותה לא רוצים לשנות. בעזרת הווקטור המחושב, ניתן להתקדם בכיוון שישפיע רק על התקדמות לכיוון המישור של הפיצ'ר שאותו כן רוצים לשנות ללא שינוי מאפיינים בתמונה ממישורים אחרים.



איור 27: התמונה נלקחה מ [16] והיא מראה הדגמה איך יוצרים התקדמות בכיוון חדש $n_2 - (n_1^T n_2)n_2$. מטילים את n_1 על n_2 ומחסרים את ההטלה מ n_1 בדרך זו אנחנו מתקדמים בכיוון מישור n_1 ללא התקדמות במישור n_2 .

באיור מספר 28 ניתן לראות דוגמה לאיך שינוי הדרגתי בווקטור הרעש יכול לשנות כמה פיצרים אשר נלמדו כקשורים בשכבה הסמויה. לדוגמה באיור זה מצד שמאל ניתן לראות איך שינוי הווקטור בכיוון המשפיע על גיל, משפיע גם על המין, בנוסף ניתן לראות איך בהורדת השפעת ווקטור המין ניתן לקבל שינוי רק בגיל האובייקט בתמונה. בצד ימין של האיור ניתן לראות איך התקדמות בשכבה הסמויה בכיוון ערכים של תמונות עם משקפיים משפיע על התוצרים גם בגיל, וכאשר מורידים את השינוי של הגיל ניתן לראות בשורה התחתונה שינוי של האובייקט רק באזור המשקפיים, ללא השינוי בגיל.



איור 28: התקדמות במרחב הלטנטי בכיוון מסוים, והורדת השפעת מאפיינים אשר אותם לא רצינו לשנות. התמונה נלקחה מ [16].

על מנת לקבל תמונה בעלת שינויים שנרצה מתמונת מקור, יש למצוא מיפוי לווקטור במרחב הסמוי אשר יהווה את התמונה החדשה אותה רוצים לקבל. לצורך כך ראשית יש צורך במציאת המיפוי של הערך ההתחלתי של התמונה המקורית למרחב הלטנטי, ולאחר מכן יש למצוא את כיוון ההתקדמות הרצוי במרחב הסמוי לצורך השינוי הרצוי בתמונה. ישנן כמה שיטות למציאת המיפוי של הערך ההתחלתי של התמונה לווקטור הרעש, שיטות אלו נעזרות במקודד נוסף. חלק מן השיטות לאימון מקודד שכזה נעשות לאחר אימון רשת ה GAN כפי שנעשה במאמר [21], וחלק מהשיטות יוצרות את המקודד במהלך האימון כגון מאמרים [17], [18].

נציג כדוגמה יצירת מקודד לפי מאמר [21] אשר מוצא את המיפוי בין תמונת תוצר של רשת הגנרטור לווקטור הרעש, המקודד נוצר לאחר אימון ארכיטקטורת GAN בעזרת רשת קונבולוציה המאומנת באמצעות אלגוריתם gradient descent על פונקציית המטרה אשר מנסה למזער את שגיאת השחזור של המיפוי. במאמר [21] מראים כי ניתן לשחזר את ערך וקטור הרעש Z בהנתן אפילו רק חלק מתוצר הרשת הגנרטיבי. בהצגה לכך שנדרש רק חלק מהתוצר הרשת הגנרטיבית, המחברים משתמשים

בפונקציות דגימה שונות, אשר ניתן לראות חלק מהן באיור 29. המחברים מראים כי עבור מטריצה הדוגמת חלק מתוצר התמונות של מוצא הרשת הגנרטיבית ניתן לייצר מקודד אשר בעזרתו ניתן יהיה לשחזר את ווקטור הרעש ממנו התמונה נוצרה ברשת הגנרטור. באופן מפורש עבור מטריצת דגימה A של חלק מהפיקסלים, המופעלת על תוצר הרשת הגנרטיבית G, עבור ווקטור הרעש Z. נרצה ליצור מודל מקודד אשר ישיג שגיאה מינימלית בהשוואה למול תמונת המקור שעבורה רוצים למצוא את ווקטור הכניסה לרשת הגנרטיבית. פונקציית ההפסד למקודד הנוסף תהיה.

$$\operatorname{argmin}_z = \frac{1}{2} (\|Y - AG(z)\|_{L_2})$$

באיור מספר 29 ניתן לראות דוגמא לתוצאות של שחזור ווקטור הרעש עבור תמונות מסוימות. בשורה העליונה ניתן לראות את התמונות המקוריות שנוצרו ברשת הגנרטיבית, בשורה האמצעית ניתן לראות את התמונה הדגומה אשר הוכנסה למקודד לצורך מציאת ווקטור הרעש המקורי לתמונה שיצר את התמונה הזו. בשורה התחתונה ניתן לראות תוצר של הרשת הגנרטיבית בהכנסת הווקטור רעש המשוחזר שהתקבל על ידי המקודד.



איור מספר 29: שחזור ווקטור רעש בעזרת מקודד, בתמונה ניתן לראות בשורות העליונות תמונה אשר נוצרה ברשת הגנרטיבית, באמצעות התמונה העליונה הדגומה, בתמונה התחונה ניתן לראות את ייצור התמונה על ידי הרשת הגנרטיבית כאשר התקבל ווקטור הרעש הנמצא על ידי המקודד כאשר הוא קיבל את התמונה האמצעית. האיור נלקח ממאמר [21].

SinGan 4.6 – אימון גנרטיבי על תמונה יחידה

עד כה לכל הארכיטקטורות והאלגוריתמים שהצגנו, היה צורך בלספק בשלב האימון מקבץ דוגמאות, מטרת האימון היתה לגרום לארכיטקטורה ללמוד את טקסטורות האפשרויות ממרחב הדוגמאות הנתון. בסט הדוגמאות, התמונות שהתקבלו היו בזוגות של תמונה והתמונה הרצויה לאחר המיפוי, או לחלופין סט דוגמאות משתי מקורות כאשר רוצים שהמיפוי יילמד על ידי הארכיטקטורה.

לעומת ארכיטקטורות קודמות, ארכיטקטורת SinGan [29] מקבלת תמונה יחידה בשלב האימון. הארכיטקטורה לומדת את מגוון הטקסטורות בתמונה בכמה רזולוציות בגדלי התמונה, ולומדת לייצר מגוון שינויים רצויים רק בעזרת התמונה היחידה המתקבלת באימון.

הארכיטקטורה מקבלת בקלט תמונה כלשהי, ומסוגלת לבצע מגוון שינויים בתמונה כגון סופר רזולוציה, שילוב אובייקטים חדשים בתמונה, שינוי ערכי תמונה כך שייגדלו אזורים מסויימים בתמונה, יצירת סרטון מתמונה. עבור כל השינויים המוזכרים, הארכיטקטורה לא משתנה וצורת האימון נשארת זהה.

הארכיטקטורה עובדת על אימון רשתות גנרטיביות במגוון רמות, בכל רמה מתמקדים ברמת שינוי ודיוק עבור מבנים שונה. ככל שעולים ברמות הארכיטקטורה הרשת הגנרטיבית יודעת להתמקד בלייצר שינויים עדינים יותר בתמונה. ברמת הראשונות בארכיטקטורה מתמקדים יותר בתפיסת המבנה הכללי של התמונה.

באופן שמזכיר את שיטת progressive growing אשר נסקרה בסעיף 3.4 בעבודה זו, גם בארכיטקטורה זו מגדילים את המודל של הגנרטור והדסקרימנטור בהתאמה ככל שתמקדמים באימון. ככל שעולים רמה מקפידים את רשת הגנרטור אשר תייצר את הכניסות לשכבה הבאה ברשת הגנרטורים. רשתות הגנרטור מחוברות בחיבור residual בתוספת תמונת רעש בכניסה. לכן מה שהרשת הגנרטיבית בכל שלב צריכה ללמוד בכל שלב הוא רק המיפוי לרעש לשינוי התמונה הדרוש על בסיס התמונה מהשכבה הקודמת. באופן מתמטי כל שכבה של גנרטור מבצעת

$$\bar{X}_n = G_n(Z_n + (\bar{X}_{n+1})^{\uparrow r}) + (\bar{X}_{n+1})^{\uparrow r}$$

הגדרת הסימונים:

\uparrow^r – פעולת הגדלת התמונה

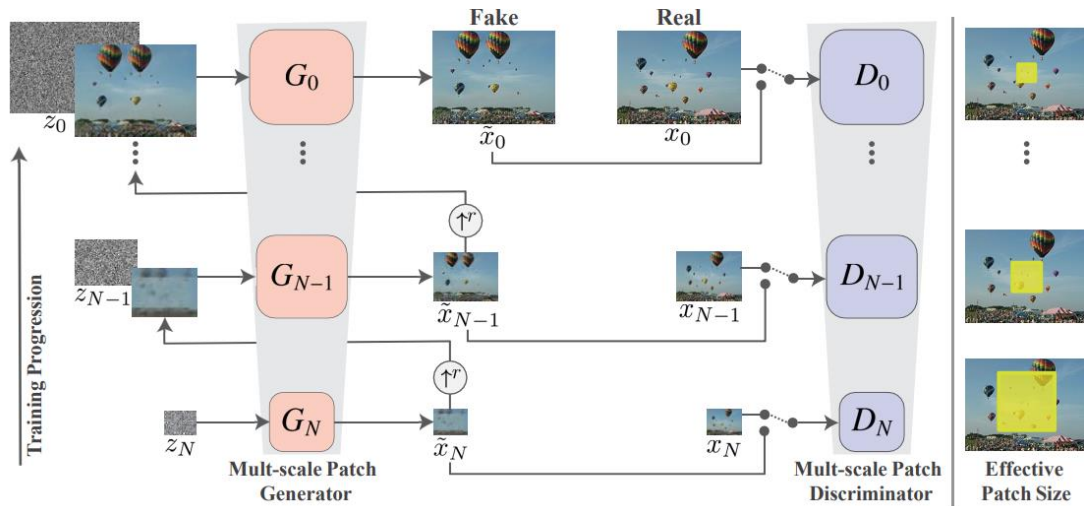
X_n – התמונה המקורית מוקטנת לגודל המתאים לשכבה ה n

\bar{X}_n – תוצר הרשת הגנרטיבית של שכבה n

נשים לב שערכי n גבוהים מייצגים שכבות נמוכות. n=0 מייצג את השכבה הגבוהה ביותר של הארכיטקטורה.

לאחר שסיימנו לאמן שכבה אין שימוש נוסף ברשת הדסקרימנטור של שכבה זו. הדסקרימנטור עובד בשיטת PatchGAN אשר תואר בסעיף 4.3 בעבודה זו, בגדלי פאטץ של 11X11 קבועים בין כל רמה לרמה. בכל רמה אנחנו מגדלים את התמונה, ומכיוון שהדסקרימנטור נשאר לסווג עם גדלי פאטץ 11X11 הוא רואה מקטעים הולכים וקטנים מהתמונה, דבר הגורם לאימון אשר מתמקד בכל רמה ברמת דיוק אובייקטים שונה, ככל שהפאטץ באופן יחסי יותר קטן בתמונה הוא מתמקד באובייקטים יותר קטנים ובטקסטורות יותר חדות. ככל שהפאטץ באופן יחסי גדול הוא מתמקד באובייקטים יותר גדולים וקווי מתאר כללים.

באיור 30 ניתן לראות את החיבורים בין השכבות, ככל שהאימון מתקדם אנחנו מגדילים את מספר השכבות, מאמנים על תמונות יותר גדולות, הפאטצים עבור הדסקרימנטור קבועים בגודלם כך שהחלק היחסי מהתמונה שהם רואים הוא קטן יותר. בכל שלב (מלבד הראשון), כניסה לרשת הגנרטיבית הוא מוצא של הרשת הגנרטיבית מהשכבה הקודמת אשר עבר הגדלה וחיבור של רעש נוסף. הרשת הגנרטיבית בכל שלב צריכה להתאים את הרעש הנוסף לאובייקטים מטקסטורטות מתאימות עבור השכבה עליה הדסקרימנטור מתאמן. שכבות הדסקרימנטור גדולות בהתאמה לשכבות ברשת הגנרטור.



איור 30: חיבור שכבות בארכיטקטורה SinGAN, בכל שכבה מגדילים את תוצר הרשת הגנרטיבית ומכניסים אותו בתוספת רעש לשכבה הבאה. כמות הפילטרים ברשת הדסקרימנטור גדלה ככל שכמות הפילטרים ברשת הגנרטיבית גדלה. האיור נלקח מ [29]

פונקציית ההפסד של הארכיטקטורה היא פונקציית הפסד עבור רשתות יריבות המשתמשת ב WGAN-GP, לפונקציית ההפסד מוספים אילוץ נוסף, תנאי שחזור, תנאי זה הוסף על מנת להכריך את המצאות התמונה האמיתית בשכבה הלנטנית.

$$L_{rec}(G) = ||G_n(0, (\bar{X}_{n+1}) \uparrow^r) - X_n||_{L2}$$

פונקציית ההפסד עבור הארכיטקטורה

$$\underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{argmax}} L(G, D) = L_{cGan}(G, D) + L_{L1}(G)$$

מחברי המאמר מראים כי גודל השינוי בתמונה מושפע מאיזה שכבת גנרטור מתחילים לייצר את התמונה. ככל שמתחילים משכבה נמוכה יותר, כך השינוי על התמונה יהיה גדול יותר.

כפי שהזכרנו, מתוצרי הארכיטקטורה ניתן לבצע מגוון משימות לשינוי ערכי תמונה. כעת נסקור את הדרכים של ביצוע השינויים לכמה משימות:

- סופר רזולוציה, כל שכבת גנרטור אשר קיבלה תמונה מוגדלת מהשכבה השניה, הוסיפה דיוק בפרטים עבור התמונה בשכבה הנוכחית. לכן כאשר אנחנו עוברים כמה פעמים בשכבה האחרונה, התהליך המתקבל הוא הוספת דיוק וחידוד לתמונה המקורית. התוצאות המתקבלות משתוות לתוצאות SRGAN אשר סקרנו בסעיף 4.1.1, כאשר ההבדל הוא ש SRGAN התאמן על מגוון גדול של תמונות בבסיס הנתונים, והארכיטקטורה הזו מתאמנת על תמונה יחידה.

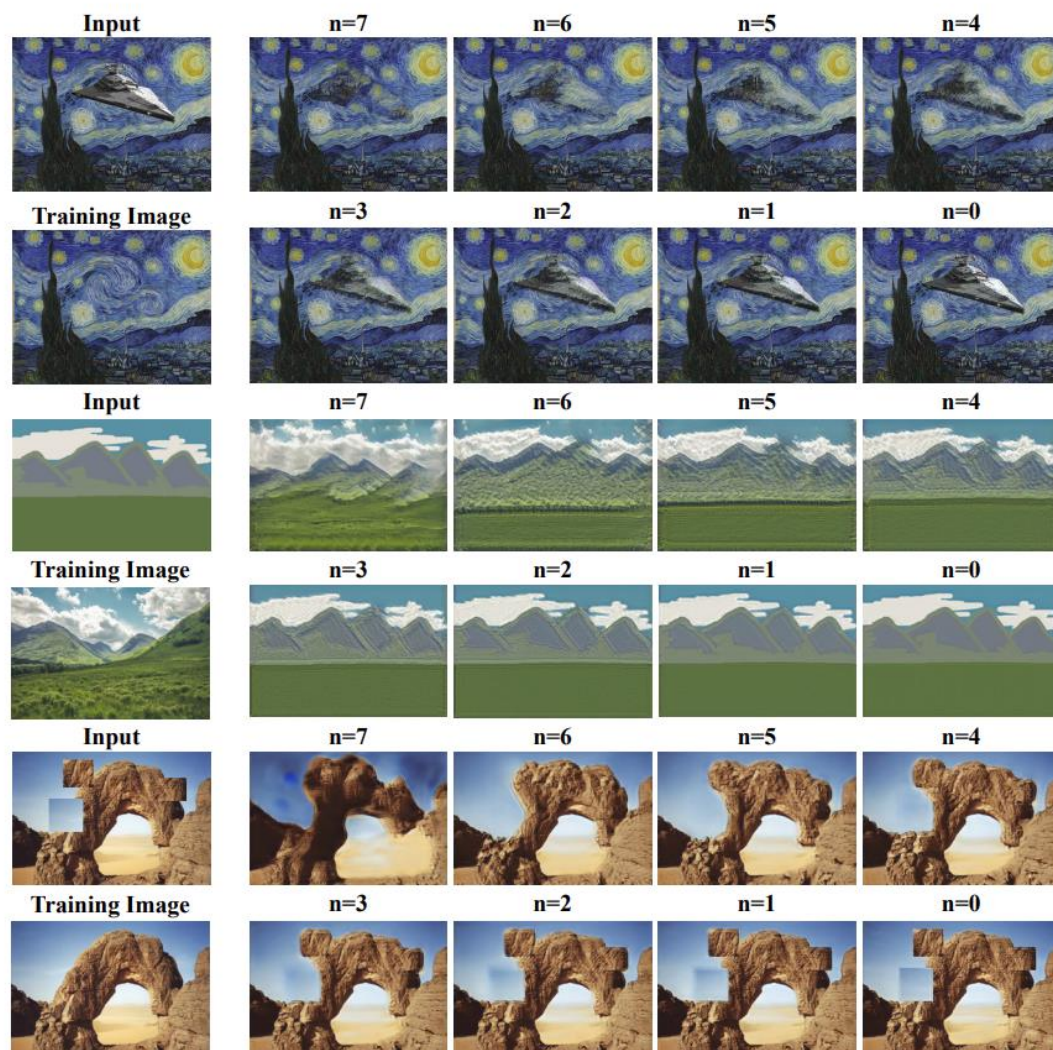
- הנפשה של תמונה, ייצור סרטון ווידאו קצר מתמונה, המחברים מראים כי הליכה על השכבה הלטנטית מייצר מעברים חלקים עבור שינוי התמונה. לכן ניתן לייצר סרטון הנפשה לתמונה מייצרת מגוון תמונות אשר השינויים בהם רציף. המחברים מדגישים שניתן לשלוט בעוצמת השינוי ככל שאנחנו מתחילים את הייצור של התמונה משכבה נמוכה יותר. כך למשל בייצור של סרטון של שייח שזז מרוח, מתמונה יחידה של שייח, המחברים מראים כי ניתן לשלוט על עוצמת הרוח על ידי הבחירה של השכבה ממנה מתחילים בייצור התמונה.

- עריכת שינויים בתמונה, על ידי חיתוך והדבקה של מקטעים בתמונה יוצרים תמונות בסיס חדשה אשר מוכנסת לארכיטקטורה, בשכבות הנמוכות. הארכיטקטורה תחליק את השינויים עבור האובייקטים במיקומים החדשים בתמונה

- הטמעת איברים חדשים בתמונה, ניתן להטמיע איברים בתמונת רקע. הארכיטקטורה צריכה להתאמן על תמונת הרקע. בשלב ההטמעה מורידים את הרזולוציה של האובייקט אותו רוצים להטמיע ומחברים לתמונת הבסיס בשכבה נמוכה של הארכיטקטורה. הארכיטקטורה תמזג את האובייקט לרקע אליה היא התאמנה בו. המחברים מראים כי ניתן לשלוט על מידת הטמעת האובייקט ככל שמתחילים משכבות נמוכות יותר.

- הפיכת ציור לתמונה, ניתן לייצר ציור אשר מהווה שינוי של התמונה, להעביר אותו בשכבות הנמוכות של הארכיטקטורה, ולקבל במוצא השכבות העליונות, תמונה אשר הבסיס של האובייקטים בה הם במבני האובייקטים מהציור, והטקסטורות של האובייקטים הם דומות לטקסטורה עליה הארכיטקטורה התאמנה עליה.

באיור 31 ניתן לראות מגוון משימות לשינוי התמונה אשר נעשה בעזרת הארכיטקטורה. בעמודה השמאלית ניתן לראות את הזוגות של התמונה עליה הארכיטקטורה התאמנה והתמונה אשר התקבל בשלב ייצרית התמונה החדשה. בעמודה השמאלית ניתן לראות את תוצרי הרשת הגנרטיבית עבור הכניסה הנתונה, כאשר עבור כל תמונה יוצרים משכבה ברמה אחרת. ניתן לראות ככל שמבצעים את יצרית התמונה החדשה מכניסות נמוכות יותר, השינויים מתמזגים יותר עם רקע התמונה. המשימות אשר ניתן לראות בתמונה הם ייצור שינוי בתמונה (2 שורות תחתונות), ייצור תמונה מציור (2 שורות אמצעיות), הטמעת אובייקט ברקע תמונה (2 שורות עליונות)



איור 31: הצגת מגוון משימות אשר אפשר לבצע בארכיטקטורת SinGan. בעמודה השמאלית מוצג התמונה עליה התאמנו והתמונה ממנה יצרו את התמונות החדשות שנמצאות בעמודה הימנית. התמונות הנמצאות בעמודה הימנית הם תוצרים שהתקבלו מהארכיטקטורה כאשר התחילו את ייצור התמונה משכבות שונות. ניתן לראות את ההשפעה של השכבה ממנה מתחילים לייצר את התמונה החדשה. האיור נלקח מ [29]

5 הערכת תוצאות רשתות ה GAN

ברשתות GAN אשר מטרתן לייצר תמונה, ניתן להשתמש במדדים לאיכות שחזור תמונה לצורך מציאת המקומות עליהן הרשת כושלת. נשים לב שמדדים אשר מנסים למדוד את כמות השינוי בפיקסלים בין תמונת המקור לתמונת המוצא מרשת הגנרטור כגון SSIM או PSNR יתנו תוצאות נמוכות בהפעלתם על תוצרי רשת GAN, זאת מכיוון שמטרת ה adversarial loss לייצר תמונה עליה רשת הדיסקרימינטור תחזיר הסתברות גבוהה לכך שהתוצר הגיע מהעולם האמיתי. עקרון זה גורם לרשת הגנרטיבית לייצר במקרי הצורך השלמה של המידע לפי העקרונות מהמרחב הסמוי אשר אותו היא

למדה, זאת בניגוד לרשתות המשתמשות במזעור ממוצע השגיאה MSE, אשר ישלימו את המידע החסר באמצעות ממוצע של האפשרויות. המדדים אשר מתמשים בחישוב מרחק פיקסלים בין תמונות בדרך כלל הם קלים לחישוב, ולכן למרות שהם לא מוצלחים במדידת תוצרים של רשתות GAN, לרוב הם יופיעו במאמרים.

מדדים מתקדמים אחרים, להשוואת ביצועים של רשתות GAN לרוב מנסים לבדוק כמה התמונה המיוצרת נראית כאילו הגיעה מהעולם האמיתי. במדדים אלה משתמשים ברשתות קלסיפיקציה או סגמנטציה שאומנו לזהות איברים בתמונה מהעולם האמיתי. ככל שהרשת מצליחה לייצר תמונות דומות יותר לעולם האמיתי עם אובייקטים ברורים, כך תוצאת המדדים תהיה טובה יותר. להלן חלק מהמדדים אשר משתמשים בהם במאמרים להערכת טיב ייצור התמונה ברשת הגנרטיבית:

PSNR 5.1

Peak signal-to-noise ratio הוא מדד אשר מודד את היחס המרבי בין האות לבין הרעש, הוא נמדד בדציבלים. בהינתן תמונה I ותוצר של תמונה משופרת K ניתן למדוד את הטעות כהפרש הערך הריבועי של כל פיקסל בתמונה.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i, j) - K(i, j)|^2$$

$$PSNR = 10 \log \left(\frac{MAX_{ij}^2}{MSE} \right)$$

כאשר MAX_{ij} הוא ערך הפיקסל הגבוה ביותר בתמונה.

החיסרון במדד זה הוא בכך שהוא לא מעניש עבור שינויים המושפעים שונה עבור התפישה האנושית, למשל עבור דוגמא בה התמונה הוזזה בפיקסל אחד נקבל ערכים נמוכים, אפילו שבעיין האנושית לא נוכל להבחין בהבדל, זאת מכיוון שכל פיקסל עכשיו ישווה לפיקסל שלא מתאים לו מבחינת מיקום. באיור 32 ניתן לראות מגוון תמונות אשר בכולם ערך ה MSE שווה, למרות שסוגי הרעש שונים, והערכת האיכות עבור עיין אנושית שונה עבור התמונות.

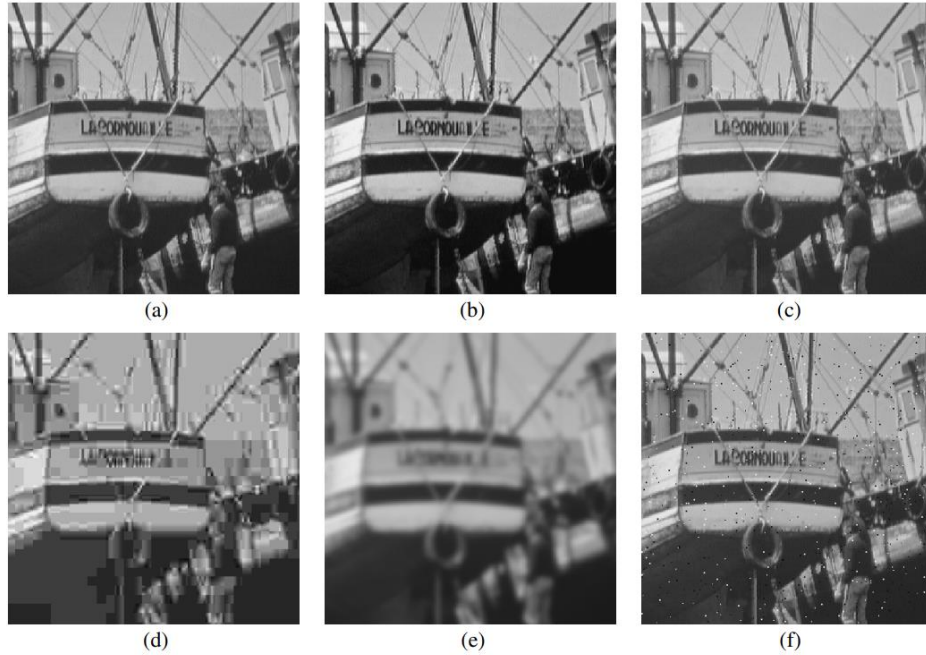


Fig. 2. Comparison of "Boat" images with different types of distortions, all with MSE = 210. (a) Original image (8 bits/pixel; cropped from 512 × 512 to 256 × 256 for visibility). (b) Contrast-stretched image, MSSIM = 0.9168. (c) Mean-shifted image, MSSIM = 0.9900. (d) JPEG compressed image, MSSIM = 0.6949. (e) Blurred image, MSSIM = 0.7052. (f) Salt-pepper impulsive noise contaminated image, MSSIM = 0.7748.

איור 32: הצגת תמונה אשר מופעל עליה כמה סוגי רעש, חישוב ערכי MSE עבור התמונה המקורית והתמונה הנוצרת מקבלים אותה תוצאה עבור כל התמונות הנוצרות. האיור נלקח מ [19]

SSIM 5.2

Structural similarity [19] הוא מדד של ערכים סטטיסטיים הבוחן אזורים בתמונה. המדד מנסה להעריך תפיסה של מבנה התמונה על ידי חלוקה לאזורים והשוואה של מבני האובייקטים בתמונה התאורה, והניגודיות.

המדד סימטרי, כלומר $SSIM(f,g) = SSIM(G,f)$

תוצאות המדד הם בין 1 ל 0, כאשר ככל שהתמונות דומות יותר יתקבלו ערכים קרובים ל 1.

עבור כמות הארה הממוצעת בתמונה ניתן לחשב את ערך הממוצע של הפיקסלים על ידי:

$$\mu_x = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{ij}$$

ניתן לחשב את היחס בהארה ממוצע התמונות:

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

נשים לב שעבור ערכי ממוצע הארה שונים באופן משמעותי נקבל תוצאות קטנות יותר ועבור ערכי ממוצע שווים נקבל ערך השווה ל 1.

משתנה C_1 נוסף על מנת להמנע מחלוקה ב 0, המשתנה הוא פרמטר לאלגוריתם וארכו צריך להיות קטן.

ניתן לחשב את היחס בניגודיות מערכי סטיית התקן בתמונות:

$$\sigma_x = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (x_{ij} - \mu_x)$$

מידת היחס בין ערכי הניגודיות:

$$C(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

נשים לב שעבור ערכי סטיית תקן שונים באופן משמעותי נקבל תוצאות קטנות יותר ועבור ערכי סטיית תקן קרובים יותר נקבל ערך המתקרב ל 1.

משתנה C_2 נוסף על מנת להמנע מחלוקה ב 0, המשתנה הוא פרמטר לאלגוריתם וארכו צריך להיות קטן.

ניתן לחשב את מבנה התמונה מהורדת הממוצע וחלוקה בשונות עבור כל תמונה $(x - \mu_x)/\sigma_x$.

נשים לב שהיחס בין $(x - \mu_x)/\sigma_x$ ו $(y - \mu_y)/\sigma_y$ הוא מקדם המתאם

$$\frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (x_{ij} - \mu_x) (y_{ij} - \mu_y)$$

מידת השוני בין מבני התמונה מחושב על ידי :

$$S(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3}$$

פרמטר σ_{xy} הוא מקדם המתאם בין x ל y , ככל שהתמונות קרובות אחת לשניה יהיה להם קורלציה גבוהה והערך יהיה קרוב ל 1 וכאשר לא קיימת קורלציה הערך של מקדם המתאם ייתקרב ל 0. משתנה C_3 נוסף על מנת להמנע מחלוקה ב 0, הוא פרמטר לאלגוריתם וארכו צריך להיות קטן

מדד SSIM הוא פונקציה של משקול של ערכי המדדים של השוני בהארה, בניגודיות, ובמבנה:

$$SSIM(X, Y) = l(x, y)^a C(x, y)^z S(x, y)^m$$

כאשר a, z, m אלו פרמטרים שניתן לשחק איתם למשקול ככל שחלק מסוים במדד יותר מעניין מאחרים.

$$MSSIM(X, Y) = \frac{1}{M} \sum_{i=0}^{m-1} (SSIM(x_i, y_i))$$

באיור 32 ניתן לראות את תוצאות ערכי MSSIM עבור תמונה שעברה מגוון שינויי רעש, ניתן לראות שערכי MSSIM הגבוהים יותר מתקבלים כאשר התמונה דומה יותר לתלמונת המקור המוצגת בתמונה השמאלית עליונה.

5.3 שימוש ברשתות נוספת למדידת גיוון ודיוק התוצרים

כפי שתיארנו עד כה, מדדים להשוואה בין תמונות מקור לתמונות הנוצרות מרשת GAN, אשר מודדים מרחק בין פיקסלים יקבלו ערכים נמוכים בהשוואה לרשתות המנסות למזער את השגיאה האפשרית בייצור התמונה. הסיבה לכך היא שמטרת רשת GAN לייצר דוגמאות הנראות חדות, דבר שיכול לבוא על חשבון שחזור מדויק של תמונת המקור. לכן לצורך מדידת טיב התמונה הנוצרת על ידי הגנרטור, נרצה להבין כמה התמונה נראית אמיתית, בנוסף נרצה למדוד את גיוון האפשרויות של יצירת התמונות. לצורך מדידה אוטומטית של איכות תמונה נוכל להיעזר ברשתות נוספות אשר אומנו על תמונות בתחום שאותו רצינו לייצר, ולשאול עד כמה הרשת מזהה אוברייקטים של תמונות שייצרנו. ככל שיהיה יותר זיהוי למאפיינים ואוברייקטים שייצרנו על רשת שאומנה לזהות אוברייקטים אמיתיים, נקבל שהתוצרים של הרשת הגנרטיבית נראים אמיתיים וחדים. נציין שיש צורך בכך שהרשת הנוספת בה נעזרים לקבלת המטריקות, יאומנו על התוצרים מהתחום שרשת ה-GAN מנסה לייצר. זאת מכיוון שכאשר נרצה להשתמש בתוצאות הרשת, או במשקולות שהיא למדה לאפיין בעזרתם פיצרים מסויימים, נרצה שיתאימו למשימה הנדרשת. לדוגמא, לא יעזור להשתמש ברשת קלסיפיקציה שאומנה להפריד על דאטה של סוגי חתולים על מנת לתת מדדי ציון לרשת GAN אשר אומנה לייצר ספרות. בעזרת רשת נוספת אשר אומנה לזהות אוברייקטים מהעולם האמיתי ניתן למדוד בכמה שיטות את טיב תוצרי הרשת הגנרטיבית שלנו, להלן כמה מהן:

5.3.1 מדידת טיב זיהוי בעזרת רשתות לזיהוי עצמים

כפי שנעשה ב [8] (מתואר בחלק של DeblurGAN:) ניתן לבחון תוצר של תמונה משפורה על ידי הפעלת רשת לזיהוי עצמים ומדידת ערכים סטטיסטיים כמו $F1$ precision recall. ככל שה precision יהיה גבוה יותר נקבל כי ערכי העצמים המזהים ברשת הדיטקציה הם הערכים האמיתיים, ככל שה-recall יהיה גבוה יותר נקבל כי העצמים המזהים מכל מחלקה זהו בתדירות יותר גבוהה. נקבל כי מטרת הערכים של המדדים precision, recall הפוכים שכן אם מזהים בצורה שקרית בכל נקודה עצם מסוג מסוים ה-recall שלי יהיה גבוה ו precision נמוך. ובצורה הפוכה, אם לא מזהים שום עצם, גם לא טועים וה-precision יהיה גבוה וה-recall נמוך. לכן עדיף במקרים אלה להשתמש במדד $F1$ שהוא הממוצע ההרמוני של שני המדדים precision, recall.

5.3.2 FCN score

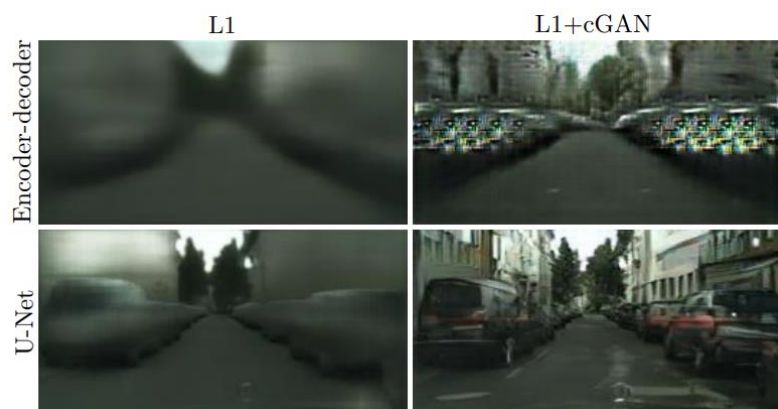
מדד בו משתמשים ב Pix2Pix I cycleGan, במדד זה משתמשים ברשת סגמנטציה אשר מתייגת כל פיקסל למחלקה שלו, הרשת מתוארת במאמר [26]. רשת הסגמנטציה פועלת לאחר אימון רשת ה-GAN, והיא צריכה להיות מאומנת על אוברייקטים מעולם שהרשת GAN אמורה לייצר. מפעילים סגמנטציה על התוצרים של רשת ה-GAN ועל תמונות אמיתיות, ולאחר מכן משווים כמות חפיפה בפיקסלים של התוויות מ-2 משימות הסגמנטציה, מדד לכמות החפיפה הוא ממוצע ה IOU בין הפיקסלים והמחלקות, כמות הדיוק הפיקסלים באופן כללי (כמות הפיקסלים הנכונים לחלק לכמות

הפיקסלים הכללית), כמות דיוק הפיקסלים בין המחלקות (כמות הפיקסלים הנכונים על המחלקה לחלק לכמות הפיקסלים הכללית במחלקה)

הרעיון מאחורי שיטה זו שאם רשת ה-GAN מייצרת תמונת ריאליסטיות התואמות לתמונת המקור אזי התוצר של הסגמנטציה צריך להיות זהה.

במאמרים Pix2Pix, CycleGan המחברים משתמים הרבה בהוכחת נכונות הבחירות שהם ביצעו בפונקציית ההפסד והארכיטקטורה בעזרת מדד FCN, להלן כמה דוגמאות אשר מחברי המאמרים מבצעים השוואה בעזרת מדד FCN:

במאמר Pix2Pix מחברי המאמר מראים כי שימוש ברשת ה-UNET כגנרטור היא עדיפה על פני שימוש בארכיטקטורת Encode Decoder. ניתן לראות כי בטבלה [1] ישנה התאמה בין איכות תוצרי התמונה למדדים של FCN SCORE, הערך הגבוה ביותר הוא עבור גנרטור מסוג UNET ופונקציית הפסד אשר תוארה בסעיף 4.3 בעבודה זו $L_{cGAN}(G, D) + L_{L1}(G)$. ניתן לראות דוגמא לשינוי באיכות המתקבל משינוי בארכיטקטורה באיור 33. באיור 33 מוצגות 4 תמונות כאשר התמונות בעמודה הימנית משתמשות בפונקציית ההפסד השלמה שצינו, והתמונות בעמודה השמאלית משתמשות בפונקציית הפסד של מרחק פיקסלים בלבד. בשורה העליונה התמונות יוצרו על ידי ארכיטקטורת כנרטור של Encoder Decoder ובשורה התחתונה התמונות יוצרו על ידי ארכיטקטורת גנרטור של UNET. ניתן לראות שערכי התמונה לדוגמא באיור 33 תואמים לערכי טבלה 1. התמונה הטובה ביותר שהתקלה היתה זו שהשתמשה בארכיטקטורת גנרטור UNET, וזו שהשתמשה בפונקציית הפסד אשר מוסיפה את CGAN.



איור 33: הצגת השוואה של ארכיטקטורת הרשת הגנרטיבית כ-UNET מול ENCODER DECODER. והצגת השפעת הגורמים השונים בפונקציית ההפסד של

Loss	Per-pixel acc.	Per-class acc.	Class IOU
Encoder-decoder (L1)	0.35	0.12	0.08
Encoder-decoder (L1+cGAN)	0.29	0.09	0.05
U-net (L1)	0.48	0.18	0.13
U-net (L1+cGAN)	0.55	0.20	0.14

טבלה [1]: שימוש במדד FCN לצורך השוואת סוגי ארכיטקטורות שונות עבור רשת הגנרטור והשוואת פונקציית הפסד מתאימה. הטבלה נלקחה ממאמר [7]

דוגמא נוספת ניתן לראות בטבלה [2], אשר מראה התאמה לאיור [21]. באיור [21] ראינו כי תוצרי הגנרטור עבור גדלי פאטץ שונים בדסקרימינטור. ניתן לראות שאכן ישנה התאמה בין התמונה לערכי FCN. ערכי ה-FCN הגבוהים ביותר מתקבל עבור מקטעים בגודל 70X70

Discriminator receptive field	Per-pixel acc.	Per-class acc.	Class IOU
1×1	0.39	0.15	0.10
16×16	0.65	0.21	0.17
70×70	0.66	0.23	0.17
286×286	0.42	0.16	0.11

טבלה [2]: שימוש במדד FCN לצורך השוואת תוצרי הרשת הגנרטיבית כתולת בגדלי המקטעים בדסקרימינטור patchGan, איור [21] מציג דוגמא לתוצרי גנרטור שונים על גדלי מקטעים שונים של דסקרימינטור.

inception score 5.3.3

המדד הוצג במאמר [27]. במדד זה מודדים שתי תכונות רצויות, כמה ברורות התמונות הנוצרות, וכמה יש גיוון בין התמונות. ככל שערכי המטריקה גבוהים, הדבר מציינ כי התוצרים של רשת ה- GAN הם גם מגוונים וגם חדים.

במדד זה דוגמים מספר תוצרים של רשת ה- GAN, את התוצרים מעבירים לרשת סיווג inception [22] (המהווה את מקור השם). בשיטה זו מודדים את תוצאות ההסתברויות לסיווג בין המחלקות השונות. עבור מדידת הדיוק, ככל שברשת הסיווג יהיו תוצאות עם ערכים גבוהים במחלקות מסוימות התמונה תחשב יותר ברורה, כלומר נרצה עבור כל סיווג תמונה שהתפלגות תהיה מצומצמת למספר מחלקות אשר בתמונה. עבור מדידת הגיוון, נרצה כי עבור מגוון תוצאות של רשת ה- GAN נקבל מגוון תוצאות שונות למחלקות ברשת הקלסיפיקציה, נרצה כי ההתפלגות למגוון המחלקות תהיה אחידה ככל הניתן.

מכיוון שרצינו כי עבור המטרות השונות (גיוון ודיוק) יהיו התפלגויות שונות בהפעלת רשת הקלסיפיקציה, נרצה שההתפלגויות מהסיווגים יהיו רחוקות. לצורך מדידת המרחק בין ההתפלגויות משתמשים במדד סטטיסטי Kullback–Leibler divergence, אשר אותו תיארו במקטע 2.1.1 בעבודה זו.

$$IS = \exp (E_{x \sim P_g} D_{KL}(P(Y|X)||P(y)))$$

מחשב ממוצע עבור כל המחלקות למדד המרחק של הגיוון מול האמינות של המחלקה.

כאשר X – תמונה נדגמת מתוצר הרשת הגנרטיבית בהתפלגות $x \sim P_g$

$P(Y|X)$ – מדד הבטחון לכל מחלקה בשכבה האחרונה של הרשת המסווגת.

$P(y)$ – הסתברות לקבלת מחלקה ברשת.

התפלגות $P(Y|X)$ מייצגת את מדד האמינות, בהינתן תמונה X המיוצרת ברשת הגנרטיבית, ככל שהיא תהיה ברורה יותר נקבל ערכים גבוהים יותר בהסתברויות למחלקה ברשת המסווגת (אנטרופיה גבוהה).

ההתפלגות $P(y)$ מייצגת את מדד הגיוון, מייצרים כמה דגימות ומעבירים לרשת הקלסיפיקציה וערך הטוב ביותר הוא כאשר יש מגוון רחב של מחלקות מיוצרות בהסתברות דומה (אנטרופיה נמוכה).

החיסרון שבשימוש ב inception score הוא שהוא מודד רק על דוגמאות שנוצרו מרשת ה GAN והוא אינו מתחשב בערכים והתפלגויות מהעולם האמיתי אשר לא נוצרו ברשת ה GAN. כך לדוגמה נגיד שהרשת הגנרטיבית מייצרת מעט איברים ברורים הניתנים להפרדה באופן מוחלט בין המחלקות, כלומר ישנה בעיה של diversity, אז $P(y)$ יהיה יוניפורמי ו $P(Y|X)$ יהיה דלתא על המחלקה x , ההסתברויות יהיו רחוקות ולכן מדד Kullback–Leibler divergence על ההתפלגויות ייתן ערך גבוה, למרות שבפועל רשת ה GAN לא מייצרת גיוון מספק.

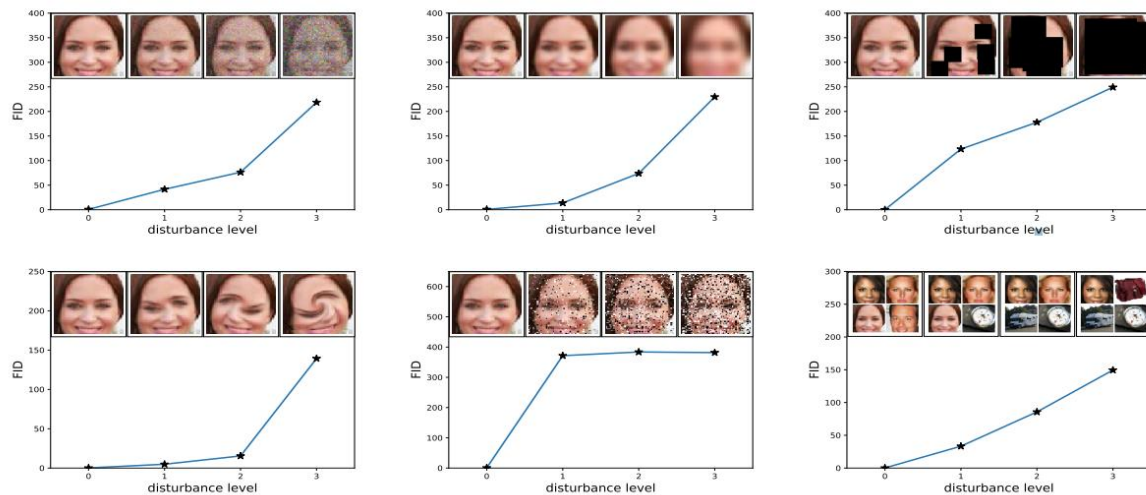
Fréchet Inception Distance (FID) 5.3.4

בניגוד למדד inception score במדד FID, אשר מוסבר במאמר [25], משווים את התפלגויות המאפיינים של האובייקטים הנוצרים בתמונת מהרשת הגנרטיבית להתפלגויות של המאפיינים של האובייקטים בתמונות מהעולם האמיתי, ולא רק את התפלגויות של תוצרי הרשת הגנרטיבית. לקבלת מרחק בין פיצ'רים של שתי תמונות משתמשים בעקרון שנקרא embedding בו מכניסים את התמונות לקלט לרשת אשר מאומנת למשימת קלסיפיקציה ומשתמשים רק בערכי השכבות הפנימיות המאומנות למציאת חלקי פיצ'רים של האובייקטים, מהשכבות הפנימיות מייצרים וקטורים של ערכים.

החישוב המתבצע במדד FID מודד את המרחק של ההתפלגויות של הווקטורים של הפיצ'רים של האובייקטים שנדגמו מהעולם האמיתי למרחק של הווקטורים של הפיצ'רים שנדגמו מהרשת הגנרטיבית, לשם נוחות מדמים את ההתפלגויות להתפלגויות נורמליות והחישוב משקלל את המרחק של ערכי הממוצע וערכי השונות המשותפת

$$FID = \left\| \mu_x - \mu_y \right\|_2^2 + TR \left(\sigma_x^2 + \sigma_y^2 - 2 \sqrt{\sigma_x^2 \sigma_y^2} \right)$$

נשים לב שהמדד מחשב מרחב התפלגויות, לכן ככל שהתפלגויות האיברים המוצגים על ידי הרשת הגנרטיבית שונות מהתפלגויות האברים בעולם האמיתי, נקבל כי המרחק יגדל, וכך גם מדד ה FID יגדל. בתמונה 34, מחברי המאמר מראים שינוי בערכי FID ככל שמוסיפים סוגי רעש שונים. בנוסף מחברי המאמר מראים כי גם עבור תמונות הנראות טוב, אבל ממקבץ תמונות שונה, מדד ה FID גדל ככל שמוסיפים יותר ממקבץ התמונות האחר.



איור 34: האיור מראה את שינוי מדד FID על סוגי רעש שונים, ככל שנוסף רעש כך מדד ה FID עולה. התמונה נלקחה ממאמר [25]

בדוגמה הימנית תחתונה ניתן לראות כי התמונות חדות, אך הם זוהמו בתמונות אשר לא מתחום תוצרי הרשת הגנרטיבית, לכן המרחק בין הפיצרים גדול ולכן תוצאות מדד FID גדלו.

במאמר SinGan אשר נסקר בסעיף 4.6 הוצא להשתמש ב SFID, מדד אשר מחברי המאמר מציעים להשתמש כאשר אין מגוון דוגמאות של תמונות אשר עליהם ניתן לקבל התפלגויות של תוצרי רשת הקלסיפיקציה. מחברי המאמר מציעים במקרה זה מדד אשר מוצא את מאפייני ההתפלגויות עבור אזורים שונים בתמונה בודדת. אופן חישוב מאפייני ההתפלגויות זהה למדד FID, ההבדל הוא שמגוון הווקטורים השונים עליהם מחשבים את מאפייני ההתפלגויות מתקבל ממגוון אזורים שונים בתמונה, במקום מווקטור יחיד עבור כל תמונה כפי שנעשה ב FID.

6. סיכום

יצירת תמונות ושינוי מאפייני תמונה בעזרת רשתות נוירונים הוא תחום מחקר גדול אשר מתפתח רבות בימים אלו. בין הארכיטקטורות הבולטות ליצירת שינויים בתמונה נפוץ השימוש בארכיטקטורות המשתמשות בעקרון של אימון רשתות יריבות (Generative Adversarial Networks (GAN)). כפי שהראנו בעבודה זו, שימוש ברשתות אלו יוצר תמונות חדות העדיפות על פני רשתות או שיטות המשתמשות בעקרון של מזעור שגיאת השחזור MSE.

בעבודה זו סקרנו מגוון אלגוריתמים שונים לשיפור ערכי תמונה אשר משתמשים בארכיטקטורת GAN כאבן בניין בסיסית בהם. שיפורי איכות התמונה אשר סקרנו היו במספר מובנים מגוונים כגון הגדלת רזולוציה, העלמת טשטוש שנובע מתזוזה, העלמת ערפל, יצירת סרטון מונפש מתמונה, עריכת תמונה, והחזרת מקטעים שנעלמו בתמונה.

בעבודה זו תיארנו את הקושי באימון רשתות GAN, המהווה את האתגר העיקרי בשימוש בשיטה. הקושי נובע ממגוון סיבות אשר אותן הסברנו, הסיבות היו קריסת מצב, העלמות הגרדיאנט באימון שנבע מבעיית המימדים הנמוכים, או מדסקרימנטור אשר מאומן בצורה עדיפה על הגנרטור כך שהגנרטור לא מקבל משוב יעיל בעדכון המשקולות. הראינו כי ישנן מגוון רחב של ארכיטקטורות אשר משפרות את יציבות האימון.

בעבודה ראינו שימוש בעיקרון Learning Transfer במאמר [10], זהו עיקרון אשר מאפשר אימון מודל עבור משימה מסויימת כאשר מתחילים ממשקולות אשר אומנו עבור משימה אחרת. עקרון Transfer Learning יכול להקטין את זמן אימון הרשת הגרטיבית על תחומים מסויימים. הרחבה לעבודה זו יכולה להיות לאיזה מבין הארכיטקטורות הנוספות עקרון Learning Transfer יכול לעזור ובאיזה סוגי תחומים ומשימות.

לרוב במשימה גנרטיבית לייצור תמונות, אין הגדרה מדוייקת איך התמונות צריכות להראות. במסגרת העבודה תיארנו מגוון מדדים לטיב איכות התוצרים של הארכיטקטורות השונות, המדדים השונים נדרשים לצורך השוואה בין סוגי תוצרי הארכיטקטורות. בין התכונות שמודדים ניתן לציין את רמת הדיוק בשחזור מבנה התמונה, רמת אמינות האובייקטים הנוצרים והגיוון של התוצרים.

בעבודה הזאת הראינו שימוש ב Patch Gan אשר זו ארכיטקטורת דסקרימנטור המסווגת מקטעים בתמונה להאם הגיעו מהעולם האמיתי או מתוצרי הרשת הגנרטיבית. הראינו שימוש לדסקרימנטור שכזה בכמה עבודות, שאלה מחקרית שאפשר לבדוק, האם יש מקרים בהם שימוש בארכיטקטורת דסקרימנטור שכזה יכולה לפגוע בתוצרים. שאלה נוספת היא האם ניתן לבחור את גודל המקטע הנכון לפי פרמטרים מסויימים במשימה או שעלינו למצוא את גודל המקטע המתאים ביותר על ידי חיפוש ממצא.

חשוב להבין את מאפייני הארכיטקטורת המגבלות והדרישות שלהן, על מנת להתאים את צורת האימון למשימה ולבסיס הנתונים אשר ברשותנו. בעבודה סקרתית ארכיטקטורת אשר משתמשת בשיטות אימון ובסיס נתונים מסוגים שונים. לדוגמא ארכיטקטורה אשר יכולה לבצע אימון על תמונה אחת בלבד ללא הצורך בבסיס נתונים, על בסיס נתונים של זוגות של תמונות אשר בוצע בהן המיפוי הרצוי, או בסיס נתונים עבור שתי מקבצי תמונות כאשר אנו רוצים שארכיטקטורה תלמד את המיפוי בין המקבצים באופן עצמאי. הרחבה מעניינת לעבודה זו יכול להיות מציאת עוד ארכיטקטורת לשינוי ערכי תמונה מבסיסי נתונים ומשימות שונות, לדוגמא מעבר בין בסיס נתונים של שמע או וידאו לבסיסי נתונים לתמונות ולהפך.

בעבודה זו סקרנו מגוון ארכיטקטורות GAN, ישנו עושר מחקרי בתחום, הנובע מהצורך למצוא דרכים לאמן את הרשת בצורה יציבה ולדייק את התוצרים למשימה אותה אנו רוצים. ניתן לצפות כי תחום מחקר זה ימשיך להתפתח ולספק תוצאות מרשימות גם בשנים הקרובות.

- [1] Goodfellow, Ian, et al. "Generative adversarial networks." *Communications of the ACM* 63.11 (2020): 139-144.
- [2] Weng, Lilian. "From gan to wgan". *arXiv preprint arXiv:1904.08994* (2019)
- [3] Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." *International conference on machine learning*. PMLR, 2017.
- [4] Arjovsky, Martin, and Léon Bottou. "Towards principled methods for training generative adversarial networks." *arXiv preprint arXiv:1701.04862* (2017).
- [5] Metz, Luke, et al. "Unrolled generative adversarial networks." *arXiv preprint arXiv:1611.02163* (2016).
- [6] Swami, Kunal, and Saikat Kumar Das. "Candy: Conditional adversarial networks based end-to-end system for single image haze removal." *2018 24th International conference on pattern recognition (ICPR)*. IEEE, 2018.
- [7] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- [8] Kupyn, Orest, et al. "Deblurgan: Blind motion deblurring using conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [9] 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.
- [10] 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.
- [11] Gulrajani, Ishaan, et al. "Improved training of wasserstein gans." *Advances in neural information processing systems* 30 (2017).
- [12] Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).
- [13] Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *Proceedings of the IEEE international conference on computer vision*. 2017.
- [14] Engin, Deniz, Anil Genç, and Hazim Kemal Ekenel. "Cycle-dehaze: Enhanced cyclegan for single image dehazing." *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018.
- [15] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434* (2015).

- [16] Shen, Yujun, et al. "Interpreting the latent space of gans for semantic face editing." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
- [17] Zhu, Jun-Yan, et al. "Generative visual manipulation on the natural image manifold." European conference on computer vision. Springer, Cham, 2016.
- [18] Donahue, Jeff, Philipp Krähenbühl, and Trevor Darrell. "Adversarial feature learning." arXiv preprint arXiv:1605.09782 (2016).
- [19] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612
- [20] Lore, Kin Gwn, Adedotun Akintayo, and Soumik Sarkar. "LLNet: A deep autoencoder approach to natural low-light image enhancement." Pattern Recognition 61 (2017): 650-662.
- [21] Ma, Fangchang, Ulas Ayaz, and Sertac Karaman. "Invertibility of convolutional generative networks from partial measurements." Advances in Neural Information Processing Systems 31 (2018).
- [22] SZEGEDY, Christian, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 1-9.
- [23] KARRAS, Tero, et al. Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196, 2017.
- [24] 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.
- [25] HEUSEL, Martin, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 2017, 30.
- [26] LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. p. 3431-3440.
- [27] SALIMANS, Tim, et al. Improved techniques for training gans. Advances in neural information processing systems, 2016, 29.
- [29] 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.

Table of content

1 Introduction	4
2 GAN architecture	6
2.1 Network loss function:.....	9
2.1.1 The optimal point of the loss function.....	11
2.2 Network training:	14
2.3 Common problems in GAN training	17
3 Improving training architectures review	23
3.1 unrolled generative adversarial networks	23
3.2 WGAN - Wasserstein GAN	25
3.3 WGAN-GP- gradient penalty	31
3.4 progressive growing of gans.....	32
4 Review leading architectures for image improvements	35
4.1 Loss function that preserve image structure	35
4.1.1 Example of use loss function that preserves image structure.....	38
4.2 CGAN - Conditional Generative Adversarial Nets.....	42
4.2.1Example of using Cgan for image improvment.....	43
4.3 Pix2Pix image translation with introduced mapping	44
4.3.1 Example of use patchGan.....	48
4.4 CycleGan architecture which learns image translation mapping	50
4.4.1 Example of use CycleGan and perceptual loss.....	52
4.5 Generating controllable changes with Latent space manipulation.	53
4.6 SinGan train on singal image	56
5 GAN products cooperation	60
5.1 PSNR.....	61
5.2 SSIM.....	62
5.3 uses of additional networks to measure diversity and fidelity	64
5.3.1 measure detection quality	64
5.3.2 FCN score	64
5.3.3 inception score.....	66
5.3.4Fréchet Inception Distance (FID)	67
6 summary.....	69
7 bibliography.....	71

abstract

Algorithms using adversarial networks are very common among image transformation tasks. The reason for this is that these algorithms manage to produce images that look real and sharp, the algorithms learn the nature of textures from real world examples during the training phase, and they manage to produce a reconstruction of these textures by completing the information as required during the activation phase.

In this work, I will explain the complexity of training adversarial networks. I will review several architectures that use the central concept of adversarial training to change image characteristics, I will focus on algorithms that improve the characteristics of the image in several diverse ways such as increasing resolution, eliminating blur, eliminating fog, creating an animated video from an image, editing an image, and returning segments that have disappeared from the image.

To understand architectures designs, I will explain the need for the changes that these architectures suggest by reviewing several advanced architectures that help the process of training adversarial networks.

In the work I will present a variety of architectures, each of which corresponds to a different type of database. Some of the architecture needs pairs of images that have a mapping from a source to the desired product. Some of the architecture can work from a pair of image files when the architecture mapping will be learned independently, another architecture that I will present will be able to learn to produce a variety of image changes from a single image without an additional database.

Algorithms that use the architecture of adversarial networks produce high-quality images, but not necessarily the closest images to the original images that you want to reproduce. There is no exact definition of how the final images in the network products should look. As part of the work, I will review a variety of algorithms and metrics for measuring the quality of the products of the various architectures, the various metrics are required for the purpose of comparing the types of architecture products.

The Open University of Israel
Department of Mathematics and Computer Science

GAN-based image quality improvement algorithms

Final Paper submitted as partial fulfillment of the requirements
towards an M.Sc. degree in Computer Science

The Open University of Israel
Department of Mathematics and Computer Science

By

Lior Bomwurzel

Prepared under the supervision of Dr. Mireille Avigal

February 2023