הפקולטה להנדסה

המעבדה לעיבוד אותות

Face Reconstruction from Voice using Generative Adversarial Networks

עידו שר שלום

עילי אהרונוביץ

פרויקט שנה ד' לקראת תואר ראשון בהנדסה

מנחה: מר יוחאי ימיני

מנחה אקדמי: פרופ' שרון גנות

אוקטובר 2022

תקציר

יצירת פרופיל אדם על פי קול הוא תחום רחב בעל מטרות רבות. על ידי שמיעת הקול של בן אדם ניתן ללמוד על המאפיינים הביו-פיזיקליים שלו, לדוגמה מה המין שלו, מה הגיל שלו, מהי מדינת המוצא ועוד.
בפרויקט שלנו נרצה לענות על השאלה: בהינתן הקלטת דובר אשר לא נראה מראש, האם נוכל לצייר פנים שיש להן כמה שיותר אלמנטים משותפים, או אסוציאציות עם הדובר, מבחינת זהות?
כדי לענות על שאלה זו נממש מערכת חישובית, פשוטה אך יעילה, המבוססת על gener-ative adversarial networks (GANs), נעשה זאת באמצעות מאמר הדן בבעיה זו בדיוק. הרשת לומדת לייצר תמונת פרצוף על ידי קטע קול של דובר באמצעות התאמה של זהויות הפנים שנוצרו לאלו של הדוברים, על סט האימון.
התוצאות מראות כי המודל מסוגל להפיק תמונות פרצוף סינטטיות המתאימות למאפיינים ביומטרים של הדובר. דיוק התוצאות איננו מקרי וטוב בהרבה מאשר ייצור אקראי של תמונה דבר המעיד על למידה של המודל.

תוכן העניינים

1 תיאור הבעיה ופתרון כללי 5

I רקע תיאורטי 5

2 Discriminative vs. Generative model 5

3 Discriminative model 6

4 Generative model 6

5 Conditional Generative Model 7

6 Explicit and Implicit density 7

7 Generative Adversarial Networks (GANs) 8

8 GANs Training Steps 9

9 Training Objective 10

10 Training Algorithm 11

11 Nash Equilibrium 11

12 Vanishing Gradient Problem 14

13 Mel Spectrogram 15

II הבעיה ופתרונה 16

14 תיאור הבעיה 16

15 תיאור הפתרון 16

III מבנה המערכת 17

16 תיאור כללי 17

17 ארכיטקטורת הרשתות 18

17.1 Voice Embedding Network 18

17.2 Generator Network 20

17.3 Discriminator and Classifier Networks 21

IV יישום ומימוש 23

18 datasets 23
18.1 audio dataset 23
18.2 vision dataset 23

19 data pre-processing 23

20 אלגוריתם האימון 24

V תוצאות 26

21 Discriminator Loss And Accuracy 26

22 Classifier Loss 27

23 Generator Loss And Output 28

VI מסקנות והצעות לשיפור 31

24 מסקנות 31

25 הצעות לשיפור 32

1 תיאור הבעיה ופתרון כללי

בפרויקט זה נעסוק בתת בעיה של יצירת פרופיל אדם על פי קול – הפקת פניו של אדם על פי קולו. ראשית, על מנת שיהיה טעם לעסוק בבעיה זו צריך לראות האם יש קשר בין הקול של אדם לתווי הפנים שלו והתשובה היא ללא ספק שכן. מחקרים נוירו-קוגניטיביים הוכיחו כי התפיסה האנושית מקשרת בין הקול של האדם לבין תווי הפנים שלו, בנוסף נערכו מחקרים בהם הראו איך בני אדם מצליחים לשפר את היכולת שלהם לקשר בין הקול לפנים כאשר הראו להם תמונות של פנים של אנשים ואת הקול שלהם. למעשה, יש אלמנטים בפנים של האדם אשר משפיעים בצורה ישירה על מערכת הקול והצלילים של האדם, למשל מבנה הפנים, ישנם אלמנטים המשפיעים בצורה עקיפה גם על תווי הפנים וגם על הקול, למשל מין, גיל, מוצא אתני ועוד. עם זאת, להפיק תווי פנים בעזרת הקול זו בעיה קשה מכמה סיבות. סיבה אחת הינה שלא ברור אילו מאפיינים בקול משפיעים בדיוק על מבנה הפנים, ולכן יהיה קשה לדעת בין מה למה יש קשר מדויק. סיבה נוספת הינה שלפעמים יהיה ניתן להבחין בהבדלים בתווי הפנים רק בצלילים מסוימים, ואם הקובץ שמע אשר מכיל את קולו של האדם לא ארוך מספיק או מונוטוני, יש חשש שלא נוכל למצוא הבדלים מספקים בין תווי פנים שונים שנרצה ליצור. בנוסף, ישנם תכונות בפנים שלא משפיעות על הקול, למשל צבע עיניים, כתמי לידה, צבע שיער ועוד.
בפרויקט שלנו, אנו מנסים לעשות שימוש בקול של בן אדם על מנת ליצור תמונה של פרצופו. לצורך פתירת הבעיה אנחנו מנסים לבדוק האם כאשר נקבל קובץ שמע שבו האדם מדבר נוכל לייצר את תווי הפנים שלו. כדי לתת מענה לבעיה זו אנו משתמשים במודל גנרטיבי הנקרא GAN, שעליו נרחיב בהמשך. במסגרת המימוש, נבנה מערכת כוללת שתפקידה יהיה לייצר תמונה המתארת פנים של אדם על סמך קבצי שמע של הקול שלו.

חלק I
רקע תיאורטי

בחלק זה נסביר מושגים בסיסים בהם נעשה שימוש לאורך הפרויקט.

2 Discriminative vs. Generative model

ניתן לסווג מודלים בלמידת מכונה לשתי קבוצות - מודלים דיסקרימינטיביים וגנרטיביים.
במילים פשוטות, מודל דיסקימינטיבי מבצע פרדיקציות על דאטא שהוא לא ראה על פי הסתברות מותנת, מתרכז בקו ההפרדה ויכול לשמש לבעיות קלסיפיקציה או רגרסיה. לעומת זאת, מודל גנרטיבי מתרכז בהתפלגות הדאטאסט, לומד מודל אשר מיצר דאטא ומחזיר הסתברות עבור כל דוגמה.

3 Discriminative model

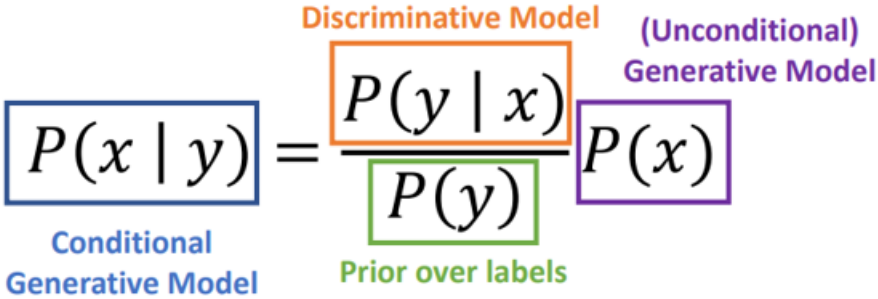
מודלים דיסקרימינטיביים מתייחסים למחלקה של מודלים סטטיסטיים המשמשים בבעיות סיווג, בעיקר משתמשים בהם בלמידת מכונה מפוקחת, supervised machine learning.
מודלים כאלו ידועים גם כמודלים מותנים מאחר והם לומדים קווי הפרדה בין לייבלים או קלאסים בדאטאסט. תפקידם הוא ללמוד התפלגות $P(Y|X)$ וכך לבצע מיפוי של משתנה כניסה אשר לא נראה מראש X ללייבל או קלאס Y התלוי בדוגמאות אימון הנצפות מראש.
החסרונות העיקריים של המודל הינם: בזמן אימון, נדרש תיוג של הלייבלים, שכן מדובר בלמידה מפוקחת, supervised settings. כמו כן, התפלגות הכניסה למשתנה, כלומר $P(X)$ אינה ידועה. ולכן, לא ניתן לדגום התפלגות זו וליצור דוגמאות חדשות של הדאטא.

4 Generative model

מודלים גנרטיביים נחשבים למחלקה של מודלים סטטיסטיים אשר יכולים לייצר דוגמאות חדשות שנוצרות מהתפלגות זהה לזו של דוגמאות האימון.
בעיקר משתמשים בהם בלמידת מכונה לא מפוקחת, unsupervised machine learning.
מודל גנרטיבי לומד את ההתפלגות המשותפת $P(X,Y)$, בפרט מהתפלגות זו ניתן לגזור את ההתפלגות השולית $P(X)$, כך אפשר לדגום את ההתפלגות וליצור דוגמאות חדשות, סינטטיות של הדאטא.
חשוב להבין כי במודל דיסקרימינטיבי אין "תחרות" בין דוגמאות שונות מהדאטאסט, ישנה "תחרות" בין לייבלים שונים אשר פוטנציאלית יכלו להיות הלייבל המתאים עבור דוגמה כלשהי.
במודל גנרטיבי לכל דוגמה, בין אם מהדאטאסט או לא, המודל יקצה הסתברות עד כמה הדוגמה קיימת.
כיוון שסכום פונקציית הצפיפות על כל דוגמאות הכניסה האפשריות הוא 1, במודל הגנרטיבי דוגמאות כניסה שונות "מתחרות" ביניהן על מסת ההסתברות.

5 Conditional Generative Model

במודל גנרטיבי מותנה לכל לייבל Y המודל לומד התפלגות עבור כל דוגמה X. כלומר, כל לייבל משרה התפלגות שונה עבור כל דוגמת כניסה.
מודל זה מחייב תיוג של הדאטא, כלומר נשתמש ב- supervised learning בלימוד של מודל גנרטיבי מותנה.
חשוב לציין כי המודלים השונים הללו לא מובחנים לגמרי ובעזרת חוק בייס ניתן לבנות מודל גנרטיבי מותנה מהמודל הדיסקרימינטיבי, הסתברות פריורית של הלייבל והמודל הגנרטיבי כמתואר מטה.



איור 1: Relationship between Discriminative and Generative Models

6 Explicit and Implicit density

נהוג להפריד את המודלים הגנרטיביים כתלות בפונקציית צפיפות ההסתברות של המודל.

- Explicit density - מספק שיערוך פרמטרי מפורש של פונקציית צפיפות ההסתברות $q_\theta(X)$ בעלת פרמטר θ ודוגמת כניסה אשר לא נראתה מראש X.
  למשל ב- Autoregressive models.
- Implicit density - שיערוך פונקציית צפיפות ההסתברות מתבצע על ידי דגימה מהתפלגות $p_{data}(x)$ מבלי להגדירה באופן מפורש. למשל על ידי GAN.

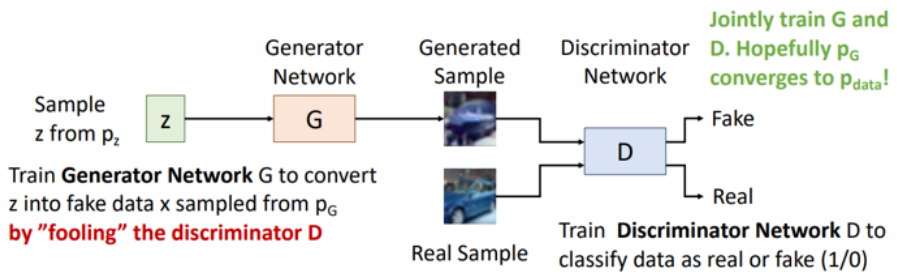
בשני המודלים מחושבת הסבירות אשר דוגמת אימון אכן שייכת לדאטאסט, השוני הוא באופן החישוב המניב את סבירות זו.
מודל אחד מבצע זאת בעזרת חישוב מפורש, explicitly של פונקציית הצפיפות. מודל שני מבצע זאת באופן מרומז, implicitly וזאת על ידי דגימה של של פונקציה אחרת שאיננה ניתנת לחישוב בצורה מפורשת.
במילים פשוטות, מודלים בעלי implicit density אינם יכולים לחשב באופן מפורש את סבירות התפלגות דוגמת הכניסה X לדאטאסט כלומר, $P(x)$, לעומת מודלים בעלי ex-plicit density אשר יכולים לחשב זאת בצורה מפורשת.

7 Generative Adversarial Networks (GANs)

GAN הוא מודל גנרטיבי בתחום למידת מכונה. במודל זה ישנן שתי רשתות נוירונים, רשת אחת הנקראת Generator ורשת שנייה Discriminator.

- מבנה: נניח כי יש לנו דאטא x_i אשר נדגם מהתפלגות $p_{data}(x_i)$.
- מטרה: ללמוד מודל אשר יאפשר לנו לדגום מתוך ההתפלגות $p_{data}(x_i)$.
- רעיון: יהי משתנה חבוי z בעל התפלגות פריורית $p(z)$. דגום $z \sim p(z)$ והעבר ברשת ה- Generator, קבל $x = G(z)$. x מתפלג מתוך p_G כלומר, התפלגות הדאטא של רשת ה- Generator.

כאשר $p_G = p_{data}$, נוכל לדגום דאטא חדש באמצעות p_G.

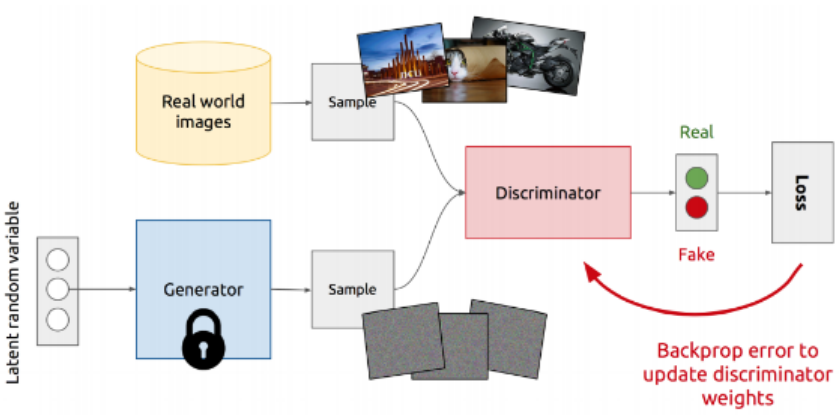


איור 2: ארכיטקטורת ה- GAN

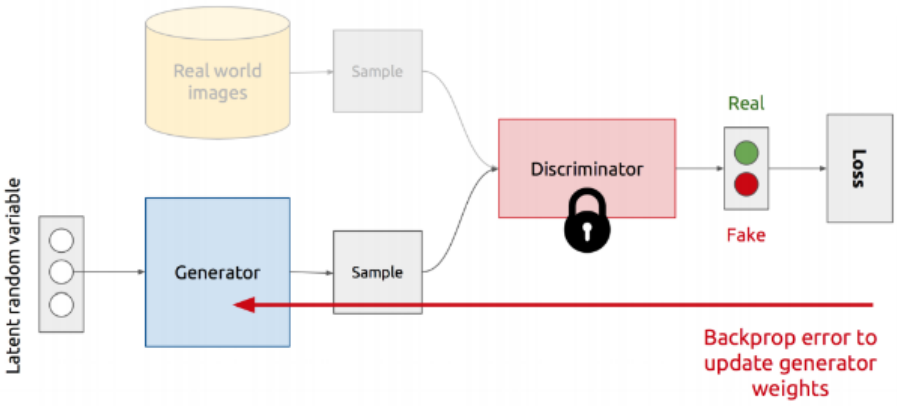
רשת ה- Generator אחראית על ייצור דוגמאות חדשות ורשת ה- Discriminator מבחינה בין דוגמאות מזויפות ומקוריות כלומר, מבצעת סיווג בינארי. במקרה שלנו, הדוגמאות הן תמונות פנים של בני אדם. מטרתנו היא לגרום לרשת ה- Generator ליצור תמונת פנים של הדובר אשר קרובה למקור, כלומר לתמונה האמיתית, ולרשת ה- Discriminator להשתפר בזיהוי, כלומר לדעת להחליט טוב יותר האם דוגמה שהיא קיבלה היא מקורית או מזויפת.
נשים לב כי לא ניתן לרשום באופן מפורש את p_G אבל, ניתן לדגום ממנו על ידי דגימה מ- $p(z)$ והעברה דרך רשת ה- Generator.
לכן ה- GAN הוא מודל מסוג implicit density, כלומר פונקציית הפילוג שלו לא ניתנת לחישוב באופן מפורש וניתן רק לדגום ממנה.

8 GANs Training Steps

מודל ה- GAN מבוסס על תחרות בין שתי רשתות, אם אחד מנצח השני מפסיד. במהלך האימון, פרמטרי הרשתות נלמדים והמודל צפוי להשתפר. שתי רשתות אלו תלויות אחת בשנייה. כאשר רשת נוירונים אחת מנצחת, נרצה לאמן את הרשת המפסידה. לדוגמה, נניח כי רשת ה- Generator מייצרת דוגמאות דאטא לא אופטימליות וה- Discriminator מסווג דוגמאות אלו כאמתיות. במצב כזה, נרצה לאמן את רשת ה- Discriminator עד אשר היא תוכל להבחין בצורה טובה בין דוגמאות מזויפות לאמתיות. אחר כך, נרצה לאמן את ה- Generator עד אשר הוא יצליח לייצר דוגמאות יותר נאמנות למקור ובכך ה- Discriminator יסווג דוגמאות אלו כאמתיות. באופן זה האימון של הרשתות מתבצע **לסירוגין** עד לנקודת שיווי משקל, Nash Equlibrium. חשוב לציין כי כאשר מאמנים רשת אחת, פרמטרי הרשת השנייה אינם משתנים.



איור 3: אימון רשת ה- Discriminator



איור 4: אימון רשת ה- Generator

9 Training Objective

כעת, ננסה להבין איך המודל עובד מבחינה מתמטית. נגדיר $G(z)$ להיות פונקציית הפלט של Generator, נגדיר $D(x)$ להיות הפונקציה המאפיינת את ה- Discriminator, מטרתה לקבוע את ההסתברות שהקלט x הוא דוגמה מה- dataset המקורי או זיוף אשר נוצר על ה- Generator. כלומר, הפלט $D(x)$ הוא הסתברות הקלט x להיות דוגמה נתונה מתוך הדאטאסט, בפרט מתקיים $0 \leq D(x) \leq 1$.
פונקציית ה- loss אשר מתארת את מודל ה- GAN הינה:

$$\min_G \max_D \left(E_{x\sim p_{data}(x)}[log(D(x))]+E_{z\sim p_{z(z)}}[log(1-D(G(z)))]\right) = \min_G \max_D V(G,D)$$

כאשר $p_z(z)$ הינה התפלגות פריורית על קלט z שהינו משתנה חבוי, latent variable של ה- Generator ו- $p_{data}(x)$ הינה התפלגות הדאטאסט.
זו פונקציית minimax שהינה פונקציית אופטימיזציה. פונקציה זו מאפשרת מודל תחרותי בין רשת ה- Discriminator לרשת ה- Generator, שהן בעלות מטרות מנוגדות. כזכור, מטרת ה- Discriminator היא לסווג את הדאטא כאמיתי או מזויף בצורה אופטימלית, מטרת ה- Generator היא לייצר דוגמאות כמה שיותר נאמנות למקור וכך לגרום ל- Discriminator לסווג דאטא מזויף כאמיתי.
כעת, נסביר מתמטית כיצד מטרות אלו תואמות את מקסום ומזעור פונקציית ה- loss בהתאמה.
נסתכל על פונקציית המחיר $V(D,G)$, נפריד לחלקים:

$$\underbrace{E_{x\sim p_{data}(x)}[log(D(x))]}_{(1)} + \underbrace{E_{z\sim p_{z(z)}}[log(1-D(G(z)))]}_{(2)}$$

(1) $x \sim p_{data}(x)$, כלומר x מפולג מתוך הדאטא האמיתי.

- ה- Discriminator שואף לסווג את x כדוגמה אמיתית דהיינו, למקסם את $D(x)$.

(2) $z \sim p_z(z)$, כלומר z מפולג מתוך התפלגות פריורית של ה- Generator.

- ה- Discriminator שואף לסווג את $G(z)$ כדוגמה מזויפת דהיינו, למזער את $D(G(z))$ באופן שקול, למקסם את $1 - D(G(z))$.

- ה- Generator ירצה כי ה- Discriminator יסווג את הדוגמאות המיוצרות על ידו כאמיתיות כלומר, ישאף למקסם את $D(G(z))$ באופן שקול, למזער את $1-D(G(z))$.

מאחר שהלוגריתם היא פונקציה מונוטונית עולה, הערך אשר ימקסם/ימזער ביטוי מתמטי מסוים ימקסם/ימזער גם את לוג הביטוי.
לכן, ה- Discriminator ירצה למקסם את $V(D,G)$ בעוד שה- Generator ירצה למזער אותו.

10 Training Algorithm

כפי שנאמר, האימון מתבצע לסירוגין בין שתי הרשתות, אלגוריתם האימון מתואר באיור 5.

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

Discriminator updates

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

Generator updates

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

איור 5: אלגוריתם אימון ה- GAN

נשאלת השאלה עבור איזו נקודה באימון נקבל את אופטימליות המודל, שאלה זו מובילה אותנו לחלק הבא.

11 Nash Equilibrium

מודל ה- GAN מבוסס על משחק סכום אפס משותף בין שני שחקנים minimax. בתורת המשחקים מודל ה- GAN צפוי להתכנס כאשר ה- Discriminator וגם ה- Generator מגיעים לנקודת שיווי משקל נאש, Nash Equilibrium. נקודת שיווי משקל נאש מתרחשת כאשר שחקן אחד לא ישנה את פעולתו ללא קשר למה שהיריב עשוי לעשות.
נקודת שיווי משקל נאש מושגת כאשר:

- $p_G(x) = p_{data}(x)\ \forall x$, התפלגות הדאטא הסינטטי זהה להתפלגות הדאטאסט.
- $D(x) = 1/2\ \forall x$, מסווג רנדומלי לחלוטין.

נקודת שיווי משקל נאש הינה המצב האופטימלי של המודל.
בנקודה זו עדכון פרמטרי הרשתות יהיה מזערי מאוד דהיינו, תהליך האימון אינו אפקטיבי יותר והמערכת במצב אופטימלי.

כעת, נוכיח מתמטית את אופטימליות המודל בנקודת שיווי משקל נאש. נעשה זאת על ידי פיתוח פונקציית המחיר של המודל.

$$\min_G \max_D V(G,D) = \min_G \max_D \left(E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_{z(z)}}\left[\log\left(1-D(G(z))\right)\right]\right) =$$

$$\min_G \max_D \left(E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{x\sim p_G}\left[\log\left(1-D(x)\right)\right]\right) =$$

$$\min_G \int_x \max_D p_{data}(x)\log D(x) + p_G(x)\log\left(1-D(x)\right)dx =$$

$$\min_G \int_x \max_D p_{data}(x)\log D(x) + p_G(x)\log\left(1-D(x)\right)dx = \circledast$$

נשים לב כי ה- Discriminator שואף למקסם את ארגומנט האינטגרל על x. נחשב את ערכו של D אשר יביא לערך מקסימלי של הביטוי.
הביטוי ניתן לייצוג על ידי פונקציה כללית. כך, נוכל לגזור את אותה פונקציה ולמצוא נקודות אקסטרמום.

$$p_{data}(x)\log D(x) + p_G(x)\log\left(1-D(x)\right) \leftrightarrow a\log y + b\log(1-y)$$

נגזרת הביטוי הינה $\frac{a}{y} - \frac{b}{1-y}$ נקבל נקודת מקסימום לוקלית עבור $y = \frac{a}{a+b}$.
ולכן, ערכו האופטימלי של ה- Discriminator הינו:

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data(x)} + p_G(x)}$$

נבחין כי לא ניתן לחשב את הביטוי בצורה מפורשת שכן ההתפלגויות $p_{data}(x), p_G(x)$ אינן ידועות באופן מפורש. נמשיך לפתח את ביטוי ה- loss, נשתמש ב-$D_G^*(x)$.

$$\circledast = \min_G \left(\int_x p_{data}(x)\log\left(\frac{p_{data}(x)}{p_{data(x)}+p_G(x)}\right) + p_G(x)\log\left(1-\frac{p_{data}(x)}{p_{data(x)}+p_G(x)}\right)dx\right) =$$

$$\min_G \left(E_{x\sim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data(x)}+p_G(x)}\right] + E_{x\sim p_G}\left[\log\frac{p_G(x)}{p_{data(x)}+p_G(x)}\right]\right) =$$

על ידי הכפלה בקבוע וחוקי לוגריתמים נגיע לביטוי הבא:

$$\min_G \left(E_{x\sim p_{data}}\left[\log\frac{2\cdot p_{data}(x)}{p_{data(x)}+p_G(x)}\right] + E_{x\sim p_G}\left[\log\frac{2\cdot p_G(x)}{p_{data(x)}+p_G(x)}\right] - \log 4\right) = \circledast\circledast$$

נזכר בשתי אנטרופיות Kullback-Leibler Divergence ו- Jensen-Shannon Divergence מתורת האינפורמציה, אשר נועדו לחישוב מרחק בין שתי התפלגויות.

$$KL(p,q) = E_{x\sim p}\left[\log \frac{p(x)}{q(x)}\right]$$

$$JSD(p,q) = \frac{1}{2}KL(p, \frac{p+q}{2}) + \frac{1}{2}KL(q, \frac{p+q}{2})$$

ניעזר בהן בחישוב הביטוי.

$$\circledast\circledast = \min_G \left(KL\left(p_{data}, \frac{p_{data}+p_G}{2}\right) + KL\left(p_G, \frac{p_{data}+p_G}{2}\right) - \log 4\right) =$$

$$\min_G \left(2 \cdot JSD(p_{data}, p_G) - \log 4\right)$$

ערך G אשר ימזער את הביטוי מתקבל עבור מציאת ערך מינימום של אנטרופיית JSD בין p_{data} ו- p_G.
ניזכר בעובדה כי JSD הוא תמיד אי שליל, ומתקבל ערך מינימלי אפס רק כאשר שתי ההתפלגויות שוות זו לזו.
לכן, ערך ה- Generator אשר ימזער את הביטוי מתקבל כאשר $p = p_{data}$.

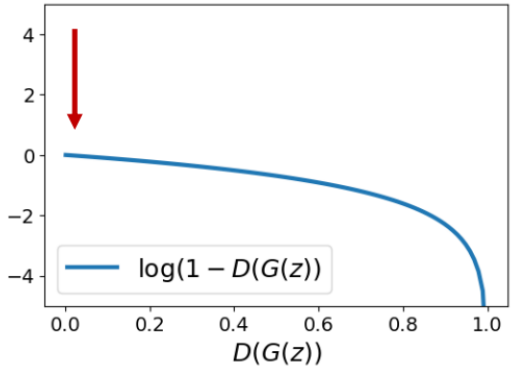
לסיכום: נקודת מינימום גלובלית של ה- minimax loss, נקודת שיווי משקל נאש מתקבלת עבור:

1. $p_G = p_{data}$, התפלגות דאטא הנוצר על ידי ה- Generator מתלכדת עם התפלגות הדאטאסט.
2. $D^*_G(x) = \frac{p_{data}(x)}{p_{data(x)}+p_G(x)}$, ערכו האופטימלי של ה- Discriminator לכל דוגמת כניסה x.

במצב זה מתקיים $D^*_G(x) = \frac{p_{data}(x)}{p_{data(x)}+p_G(x)} = \frac{1}{2}$ כלומר, הסיווג מתבצע בצורה רנדומלית.

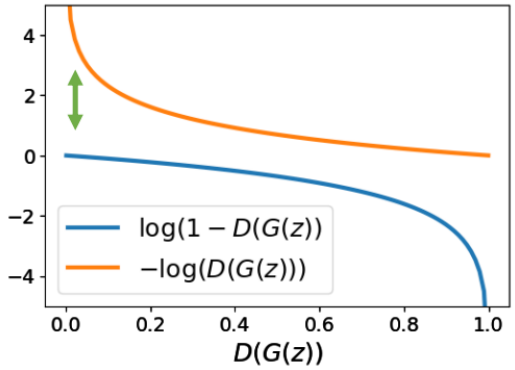
12 Vanishing Gradient Problem

נשים לב כי בתחילת אימון מודל ה- GAN רשת ה- Generator תוציא כפלט דוגמאות דאטא גרועות, זבל רנדומי. דוגמאות אלו יסווגו בקלות על ידי ה- Discriminator, סיווג דאטא אמיתי כנגד דאטא "זבל" הוא פשוט, ה- Discriminator יסווג נכונה לאחר כמה צעדי התקדמות של האלגוריתם. ולכן, בתחילת האימון $D(G(z))$ קרוב מאוד לאפס, פונקציית המחיר עבור ערך זה היא שטוחה כפי שמתואר מטה באיור 6. כתוצאה מכך, הגרדיאנט הוא אפס, צעדי ההתקדמות על העקומה הם מזעריים, כמעט ואין עדכון בפרמטרי הרשת. במילים אחרות, ה- Generator לא ילמד דבר והמודל יכשל.
בעיה זו נקראת Vanishing Gradient, היא באה לידי ביטוי באימון רשתות נוירונים עמוקות על ידי שיטות מבוססות גרדיאנטים ובפרט בעדכון פרמטרי הרשת באמצעות אלגוריתמי backpropagation. בעיית Vanishing Gradient הינה תחום מחקר פעיל עוד היום.



איור 6: Generator Loss Function

אחד הפתרונות הוא לאמן את ה- Generator למזער פונקציית loss שונה. וכך, בתחילת האימון מתקבלים ערכי גראדינט גדולים. כלומר, בעת עדכון פרמטרי רשת ה- Generator, במטרה למזער את גראדיאנט ה- loss, הפרמטרים מתעדכנים כדרוש והרשת לומדת.



איור 7: עקומות ה- loss, מניעת בעיית Vanishing Gradient בתחילת האימון

13 Mel Spectrogram

ספקטרוגרמת מל הינה ייצוג ספקטוגרמי בסקאלת מל, כדי להבין את משמעותה נצטרך ללמוד על המושגים Spectrogram ו- Mel Scale.

- Spectrogram - הוא ייצוג גרפי של תדירויות נמדדות, המשתנות לפי זמן.
ספקטרוגרמה של אות שמע נוצרת על ידי חלוקת האות לחלונות זמן, הפעלת התמרת פורייה על כל חלון זמן, לבסוף התאמת סקלת התדרים ל- Log Scale והאמפליטודה לדציבלים.

- Mel Scale - הוא תוצאת טרנספורמציה לא לינארית על סקלת התדרים. סולם מל ממדל מרחקים בין אותות דיבור, המרחק בין אותות שמע בסולם מל הוא פרופורציונלי למרחק הנוצר על ידי שמיעה של בני אדם.

ייצוג ה- Spectrogram בסקאלת ה- Mel Scale יניב את ה- Mel Spectrogram.
נשים לב כי הסקאלה שלנו היא לא לינארית כמו ספקטרוגרמה רגילה, אלא מדובר בסקאלה מיוחדת של תדירויות. ה- Mel Spectrogram ממפה את הערכים של התדירויות שמיוצגים בהרץ לערכי ה- Mel Scale, אשר פרופורציונליים להפרשי המרחק עבור האוזן האנושית.
חשוב להבין כי ב- Linear Spectrogram לכל התדירויות יש חשיבות שווה, אבל לא תמיד אנו רוצים זאת, לפעמים נרצה להבליט תדירויות מסוימות או לבדוק יותר לעומק מה קורה בהן או לחילופין לייחס פחות חשיבות לתדירויות אחרות. יש שימוש ב- Mel Spectrogram במודלים המנסים לבחון את חוש השמיעה של בני האדם, זאת מכיוון שגם בשמיעה של בני האדם אין התייחסות שווה לתדרים, ובדרך כלל בני אדם מסוגלים להבדיל טוב יותר בין תדרים נמוכים מאשר בין תדרים גבוהים.

חלק II
הבעיה ופתרונה

לאחר שהסברנו והרחבנו אודות תהליך הלמידה, מושגים, שיטות ועוד, נוכל כעת להרחיב בצורה מעמיקה על הבעיה אותה נרצה לפתור ואת פתרונה.

14 תיאור הבעיה

על מנת לפתור את הבעיה נצטרך לחשוב כיצד לבנות את ארכיטקטורת הרשתות. כלומר, איזה רשתות נוירונים יש לשים מלבד הרשתות הבסיסיות של מודל ה- GAN.
צריך לחשוב ולבדוק איזה שכבות לקחת בכל רשת, אופן סדר השכבות, אילו פונקציות אקטיבציה יש להשתמש, ערכי היפר-פרמטרים, אופן חלוקת הדאטאסט שבידינו לקבוצות (training, validation, test), גודלן של קבוצות אלו, האם כמות הדאטא מספיקה לאימון המודל, האם יש לנו מספיק משאבי זיכרון וחישוב לאימון רשתות הנוירונים וכו'.

15 תיאור הפתרון

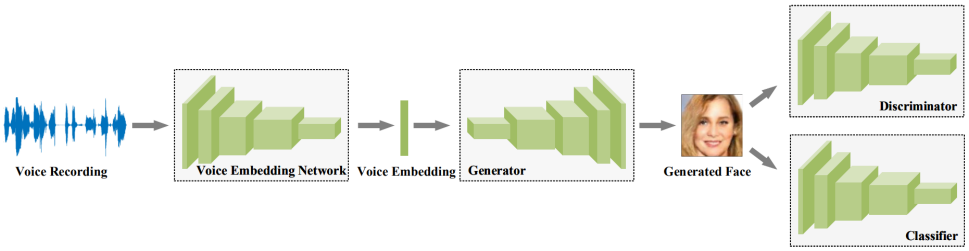
הבנת הפתרון התיאורטי נעשתה על ידי קריאת מאמרים וחומר העוסקים בפתרון בעיות אלו. הבנת הפתרון הפרקטי נעשתה לרוב על ידי ניסוי ותהייה לדוגמה, בחירת היפר-פרמטרים מתאימים אשר יביאו לתוצאות טובות ביותר. המימוש הפרקטי נעשה על ידי כתיבת קוד מתועד ומודלרי באמצעות Pytorch למימוש רשתות הנוירונים של המערכת. תחילה, עבור כל רשת הוגדרה מחלקה אשר תייצג אותה. בכל מחלקה הגדרנו את השכבות המרכיבות כל רשת שכבה אחר שכבה. נעזרנו בפונקציות מובנות אשר סייעו לנו באימון המודל. כמו כן, תהליך קריאת הדאטא, ניתוחו והכנסתו לתוך מבני נתונים מתאימים אשר יאפשרו לנו שימוש לצרכינו לטובת תהליך האימון והבחינה. לבסוף, בחנו את הדיוק והטיב של המודל על ידי בחינה של הנתונים שבידינו מול הפלט של הרשתות והוצאנו גרפים מתאימים כך שיהיה ניתן לראות ויזואלית את התוצאות.

חלק III
מבנה המערכת

בחלק זה נעמיק אודות מבנה המערכת, תתי המערכות המרכיבות אותה וזרימת הנתונים דרך המערכת כולה. נתחיל בתיאור כללי, לאחריו נפרט על כל אחת מתתי המערכות בצורה נרחבת, נסביר אודות ארכיטקטורת רשתות הנוירונים נבין את קלטי המערכת ואת תוצריה.

16 תיאור כללי

ראשית, נזכיר כי מטרתנו היא ללמוד ליצור פרצוף על סמך קול של דמות. ולכן, קלט המערכת שלנו הוא הקלטת אודיו של הדובר והפלט הוא תמונת פרצופו.
נחלק את המערכת הכוללת לתתי-מערכות עיקריות: Voice Embedding Network, Generator, Discriminator, Classifier.
מערכת ה- Voice Embedding Network נועדה לחלץ את המידע המשמעותי מהקלטה של הדובר. נרצה להעביר את קלט קובץ האודיו במערכת עיבוד כלשהי, Preprocessing . זאת על מנת לזהות מרכיבים משמעותיים בהקלטה אשר משפיעים בצורה ישירה על תווי הפנים של הדובר. לדוגמה, אם קובץ הקלט מכיל רעשי רקע נרצה לנקות אותם כדי שלא ישפיעו על דרך יצירת הפרצוף על ידי מערכת ה- Generator. פלט מערכת זאת הוא וקטור המכליל את כל האינפורמציה הבולטת של הקלטת הדובר, והוא נכנס ישירות כקלט למערכת ה- Generator. מערכת ה- Generator מייצרת תמונה של הדובר ומוציאה אותה כקלט למערכות Discriminator ו- Classifier. המערכת שלנו מבוססת GAN ותהליך האימון מבוסס על זוג Discriminators, שניהם מקבלים כקלט תמונה (אמיתית או מזויפת). הראשון, ה- Discriminator, משערך האם תמונת הקלט היא תמונת פרצוף אמיתית. השני, ה- Classifier, מסווג את זהות תמונת הקלט, מספר id של דובר כלומר, מי הדובר מתוך דאטא האימון אשר נמצא בתמונה.
איור של המערכת ורשתות הנוירונים אשר מרכיבות אותה מתואר מטה.



איור 8: המערכת המוצעת אשר מבוססת GAN ליצירת פרצוף דובר מתוך הקלטת קולו. כוללת 4 רשתות נוירונים: voice embedding network, generator, discriminator, classifier.

17 ארכיטקטורת הרשתות

בחלק זה נרצה להציג ולתאר את מבנה וארכיטקטורת רשתות הנוירונים בהן השתמשנו. כמו כן, נתאר את קשרי הרשתות, הקלטים, המוצאים שלהן ועוד. תיאור זה יעזור להבנת flow המערכת ותוצריה.

17.1 Voice Embedding Network

רשת נוירונים זו הינה אבן הבניין המרכזית של המערכת, היא הראשונה בזרימת הנתונים דרך המערכת כולה. אחראית בין היתר על ניקוי האות וקידודו.
קלט רשת נוירונים זו הינו אות דיבור אשר נדגם בתדר דגימה $f_s = 16[kHz]$, משך זמן אות זה הינו t_0 שניות.
אות הדיבור הינו הקלטת דובר לא מעובדת כלומר, לא עברה ניקויי רעש או רקע וגם אינה מקודדת. רשת זו מוציאה כפלט קידוד של אות השמע.
ביתר פירוט, הרשת מקבלת וקטור אודיו ממימד t_0 ומוציאה כפלט וקטור קידוד ממימד 64 וקטור הקידוד הוא בעצם ה- Latent random variable.
כלומר, משתנה חבוי אשר בעזרתו מערכת ה- GAN לומדת.
עיקר שכבותיה של הרשת הינם שכבות קונבולוציה חד מימדיות זהות בעלות הפרמטרים הבאים: Kernel Size=3, Stride=2, Padding=1.
לאחר כל שכבת קונבולוציה ישנה שכבת Batch Normalization ולאחריה, פונקציית אקטיבצייה Relu.

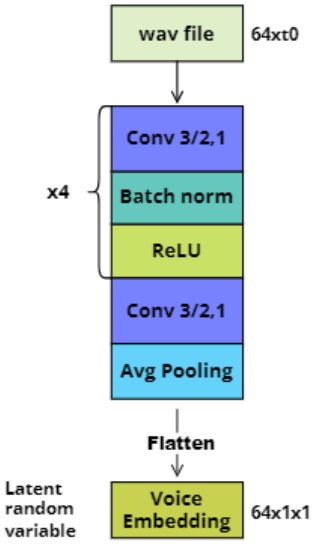
ניזכר בנוסחה הבאה לחישוב גודל מימד הפלט עבור שכבת קונבולוציה:

$$W_o = [(W_i - K + 2P)/S] + 1$$

כאשר, W_o הינו מימד הפלט, W_i הינו מימד הקלט, K הוא גודל החלון, P הוא גודל ה- Padding ו- S הוא גודל הצעד כלומר, מספר צעדי תזוזות החלון.
בעזרת נוסחה זו נוכל לחשב את פלט שכבת הקונבולוציה עבור ערכי הפרמטרים במקרה שלנו:

$$t_{i+1} = [(t_i - K + 2P)/S] + 1 = [(t_i - 3 + 2 \cdot 1)/2] + 1 \Rightarrow \boxed{t_{i+1} = [(t_i - 1)/2] + 1}$$

קיבלנו משוואה איטרטיבית המתארת את מימד טנזור הפלט t_{i+1} כתלות במימד טנזור הקלט t_i.
משוואה זו מאפשרת לנו לחשב עבור כל שכבת קונבולוציה את מימד פילטר המוצא.
כמו כן, נציין כי נלקחים כמות פילטרים, activation maps שונה בכל הפעלה של שכבת קונבולוציה דבר זה גורר חילוץ פיצ'רים בצורה מגוונת מהתמונה.
לבסוף נקבל וקטור ממימד t_5 (בהתאם ל- 5 שכבות הקונבולוציה ברשת) עבורו יתקבלו 64 activation maps,
נבצע Average Pooling בניהם על פני מימד t_5. כלומר, נמצע את הפילטרים. הפלט הסופי הינו וקטור קידוד ממימד 64.



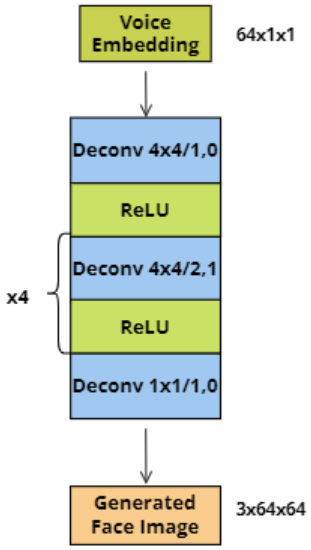
Voice Embedding Network		
Layer	Act.	Output shape
Input	-	$64 \times t_0$
Conv $3_{/2,1}$	BN + ReLU	$256 \times t_1$
Conv $3_{/2,1}$	BN + ReLU	$384 \times t_2$
Conv $3_{/2,1}$	BN + ReLU	$576 \times t_3$
Conv $3_{/2,1}$	BN + ReLU	$864 \times t_4$
Conv $3_{/2,1}$	BN + ReLU	$64 \times t_5$
AvePool $1 \times t_5$	-	64×1

איור 9: מבנה שכבות רשת הנוירונים Voice Embedding Network

כאמור, פלט שכבה זו הוא משתנה חבוי אשר נכנס כקלט לרשת הנוירונים Generator עליה נרחיב בעמוד הבא.

17.2 Generator Network

אחר מעבר ברשת ה- Voice Embedding, קיבלנו קידוד מתאים להקלטת הדובר, וקטור קידוד שמע זה הינו קלט רשת ה- Generator או במילים אחרות משתנה חבוי המאפשר את למידת התפלגות הדאטא הסינטטי, דאטא רשת ה- Generator.
פלט הרשת הוא מטריצה בעלת המימדים 3x64x64, הפלט בעצם מייצג תמונה. ביתר פירוט, קידוד כמותי של הצבעים של הפיקסלים בתמונה שנוצרת – בפורמט RGB הצבע שאנחנו רואים בפיקסל הספציפי במסך נקבע לפי החלוקה של העוצמות בשילוב בין אדום, ירוק וכחול.
הרשת מורכבת משש שכבות קונבולוציה דו מימדיות בעלות פרמטרים שונים. ניזכר כי ברשת הקודמת עבדנו עם שכבות קונבולציה חד מימדיות, וכאן אנו משתמשים בשכבות קונבולוציה דו מימדיות. כלומר, ה- Kernel אשר מחליק על הקלט הוא דו מימדי דבר זה הגיוני שכן פלט הרשת הוא תמונה דו מימדית.
אנחנו משתמשים בשכבות קונבולציה בדרך כלל כאשר אנחנו רוצים למצוא מידע כמותי סטטיסטי מתוך אזורים קרובים בדגימות שלנו, באופן זה ניתן להשיג מידע יותר אינפורמטיבי בעל קורלציה בין פיצ'רים שונים.
ברשת זו אנו צריכים לספק כפלט תמונה אשר מיוצגת בפורמט RGB כפי שהוסבר, ולכן נצטרך לעבור מקלט של וקטור חד מימדי לפלט בעל 3 מימדים. לשם כך אנחנו משתמשים בשכבות קונבולציה דו מימדית כאן.
לאורך כל הרשת, לאחר כל שכבת קונבולוציה כזו, למעט האחרונה, ישנה פונקציית אקטיבצייה Relu.
מבנה הרשת מתואר באיור מטה.



Generator		
Layer	Act.	Output shape
Input	-	$64 \times 1 \times 1$
Deconv $4 \times 4_{/1,0}$	ReLU	$1024 \times 4 \times 4$
Deconv $3 \times 3_{/2,1}$	ReLU	$512 \times 8 \times 8$
Deconv $3 \times 3_{/2,1}$	ReLU	$256 \times 16 \times 16$
Deconv $3 \times 3_{/2,1}$	ReLU	$128 \times 32 \times 32$
Deconv $3 \times 3_{/2,1}$	ReLU	$64 \times 64 \times 64$
Deconv $1 \times 1_{/1,0}$	-	$3 \times 64 \times 64$

איור 10: מבנה שכבות רשת הנוירונים Generator

17.3 Discriminator and Classifier Networks

מערכת ה- GAN הנתונה לומדת על ידי זוג Discriminators אשר יקראו Discriminator ו- Classifier שניהם מקבלים כקלט תמונת פרצוף. תמונה זו יכולה להיות אמיתית או מזויפת. במידה וקלט רשתות אלו הוא תמונה מזויפת אזי קלט זה מתקבל על ידי מעבר ברשתות Voice Embedding ו- Generator.

כלומר, מעבר מהקלטה של הדובר לקידוד של ההקלטה, משם לתמונה אשר מיוצגת על ידי מטריצה של ערכים של פיקסלים בפורמט RGB.

אחרת, הקלט הוא תמונה מקורית מתוך הדאטאסט והוא מתקבל ישירות ללא מעבר ברשתות Voice Embedding ו- Generator.

ה- Discriminator, בוחן האם תמונת הקלט היא תמונת פרצוף אמיתית או מזויפת. הוא יחזיר פלט הסתברותי, מספר בין 0 ל- 1, אשר מתאר את הסתברות תמונת הקלט להיות שייכת לדאטאסט.

ה- Classifier, מסווג את זהות תמונת הקלט. יחזיר וקטור הסתברויות של אישיות הדוברים מהדאטאסט כאשר האיבר ה- i בוקטור זה הינו הסתברות זהות תמונת הקלט להיות הדובר ה- i. לכן, פלט רשת ה- Discriminator הינו ממימד 1 ופלט רשת ה- Classifier הוא ממימד k כאשר ישנם k דוברים בדאטאסט.

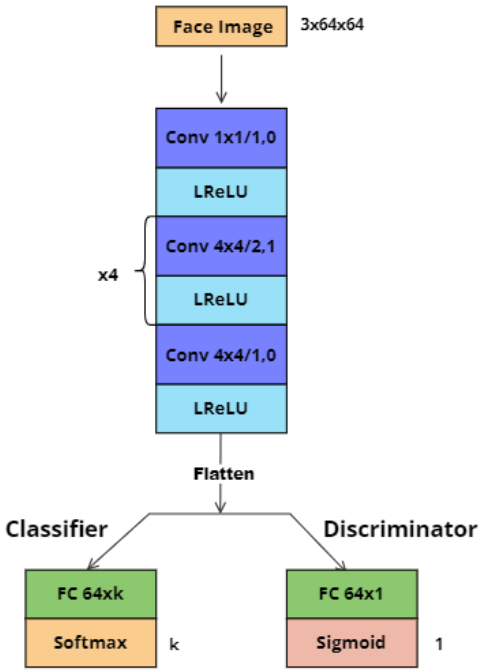
נשים לב כי זוג ה- Discriminators הללו מניבים וקטור התפלגויות אשר סכומו הוא 1. כל אחד מהם הוא מסווג רך. על ידי לקיחת הערך בעל ההסתברות הגבוהה ביותר, argmax, נקבל את הסיווג הקשה. רשתות הנוירונים Discriminator ו- Classifier הינן בעלות ארכיטקטורה זהה פרט לשתי השכבות האחרונות שם הן שונות זו מזו. ביתר פירוט, עד לשתי השכבות האחרונות לשתי הרשתות ישנן שכבות משותפות. חשוב להבין כי הפרמטרים של שכבות אלו משותפים לשתי הרשתות (shared weights).

כאשר מתבצע תהליך האימון של הרשתות פרמטרי הרשתות עוברים תהליך אופטימיזציה למזעור פונקציית ה- Loss. שתי הרשתות משנות את הערכים של הפרמטרים המשותפים לסירוגין כלומר, הרשתות משפיעות אחת על השנייה.

רשת הנוירונים המרכיבה את השכבות המשותפות תקרא Face Embedding Network והיא מורכבות בעיקר מצמד שכבות של שכבת קונבולציה דו מימדית, ולאחריה פונקציית אקטיבציה Leaky Relu, כאשר צמד שכבות זה חוזר מספר פעמים. כאמור, השוני הינו בשתי השכבות האחרונות. בסופה של רשת ה- Discriminator ישנה פונקציית אקטיבצייה Sigmoid ובסופה של רשת ה- Classifier ישנה פונקציית אקטיבצייה Softmax.

דבר זה הגיוני שכן עבור זוג ה- Discriminators מתקבלים בפלט הסתברויות. עבור ה- Classifier מתקבל **וקטור** התפלגויות ולכן, נשתמש בפונקציית אקטיבצייה Softmax. עבור ה- Discriminator מתקבל ערך הסתברות **יחיד** ולכן, נשתמש בפונקציית אקטיבצייה Sigmoid.

איור 11 מתאר את רשתות הנוירונים Discriminator ו- Classifier, מבנה השכבות שלהן, פרמטרי כל שכבה ומימדי הקלט והפלט.



Discriminator		
Layer	Act.	Output shape
Input	-	$3 \times 64 \times 64$
Conv $1 \times 1_{/1,0}$	LReLU	$32 \times 64 \times 64$
Conv $3 \times 3_{/2,1}$	LReLU	$64 \times 32 \times 32$
Conv $3 \times 3_{/2,1}$	LReLU	$128 \times 16 \times 16$
Conv $3 \times 3_{/2,1}$	LReLU	$256 \times 8 \times 8$
Conv $3 \times 3_{/2,1}$	LReLU	$512 \times 4 \times 4$
Conv $4 \times 4_{/1,0}$	LReLU	$64 \times 1 \times 1$
FC 64×1	Sigmoid	1

Classifier		
Layer	Act.	Output shape
Input	-	$3 \times 64 \times 64$
Conv $1 \times 1_{/1,0}$	LReLU	$32 \times 64 \times 64$
Conv $3 \times 3_{/2,1}$	LReLU	$64 \times 32 \times 32$
Conv $3 \times 3_{/2,1}$	LReLU	$128 \times 16 \times 16$
Conv $3 \times 3_{/2,1}$	LReLU	$256 \times 8 \times 8$
Conv $3 \times 3_{/2,1}$	LReLU	$512 \times 4 \times 4$
Conv $4 \times 4_{/1,0}$	LReLU	$64 \times 1 \times 1$
FC $64 \times k$	Softmax	k

איור 11: מבנה שכבות רשתות הנוירונים Discriminator, Classifier

חלק IV
יישום ומימוש

בחלק זה נפרט אודות היישום ומימוש הפרויקט. הדאטאסטים בהם השתמשנו, עיבוד נתונים מקדים, מימוש הקוד, סיפריות מרכזיות ועוד.

18 datasets

בפרויקט שלנו, למידת המודל מתבצעת על ידי הקלטות דוברים ותמונות פרצוף שלהם. דהיינו, ישנם שני מאגרי נתונים להם נזדקק האחד של הקלטות הדוברים והשני של תמונות הפרצוף. עבור שני הדאטאסטים הללו נלקחו חיתוך הדוברים. כלומר, לא קיימת הקלטה מדאטאסט האודיו שאין לה תמונת פרצוף מדאטאסט התמונות וכך גם הפוך.
נציין כי הדאטאסטים נלקחו מתוך מאמרים אשר מטרתם הייתה לייצר מאגרי מידע אלו, הם לא נוצרו על ידינו.

18.1 audio dataset

הקלטות הדוברים נלקחו מ- Voxceleb dataset שהוא מערך נתונים אודיו-ויזואלי המורכב מקטעים קצרים של הקלטות דוברים, הוא מופק מסרטוני ראיונות שהועלו ליוטיוב.
מערך הנתונים מורכב מ- 1,251 זהויות. לכל זהות ישנם מספר קבצי שמע.

18.2 vision dataset

תמונות הפרצוף נלקחו מ- VGGFace dataset, מערך נתונים המורכב מ- 2,622 זהויות. לכל זהות יש קובץ טקסט משויך המכיל כתובות URL לתמונות של אותה זהות ועוד מטא-דאטא.
בפרויקט שלנו, נעשה שימוש בגרסה ידנית, מסוננת של הדאטא.
ביתר פירוט, הדאטאסט המקורי מורכב מקבצי טקסט המכילים קישורי אינטרנט לתמונות של הזהות. חלק מהקישורים אינם ואלידיים והתמונות אינן בהכרח תמונות פנים.
כלומר, הדאטאסט הוא אינו מסונן. הגרסה בה אנו השתמשנו מכילה קבצי jpg של תמונות פרצוף ממורכזות עבור האישויות מהקישורים. בגרסה אחר הסינון ישנן 2,554 זהויות.

19 data pre-processing

על מנת לשפר את ביצועי המודל והמערכת נבצע עיבוד מקדים כגון, עיבוד רעשים באודיו, או נורמליזצייה של תמונות הפנים. עבור סגמנטי האודיו אנו משתמשים בממשק המזהה פעילות קולית אקטיבית. אנו עושים זאת תוך שימוש בפרויקט WebRTC כדי לבודד אזורים נושאי דיבור בהקלטות. אחר כך, מופעלת mel-spectrogram על הקלטת האודיו.
עבור דאטא התמונות, כל פיקסל מנורמל על ידי הפחתת 127.5 ואז חלוקה ב-127.5.
לאחר שסיננו את הנתונים הלא רלוונטיים ממאגרי הנתונים שבידינו, נרצה להיות עם מאגר נתונים שיכלול רק את הזהויות שעבורן יש לנו גם קבצי שמע וגם תמונות פרצוף. לצורך כך, כתבנו קטע קוד שמוצא את החיתוך בין מאגרי הנתונים השונים. היינו צריכים להשתמש

במזהה כלשהו שהוא חד ערכי, והשתמשנו בשם של האדם שהנתונים שייכים אליו (חילצנו את זה מתוך שמות התיקיות שקיבלנו). יצרנו מילון שהמפתח בו הוא שם האדם והערך הוא מיקומים של הקבצים ששייכים לו במאגר הנתונים. בתחילה, השתמשנו בספרייה בשם glob על מנת לסרוק תיקיות ותתי תיקיות על מנת לחלץ את כל הקבצים והנתונים הרלוונטיים מתוך התיקיות שקיבלנו במאגרי הנתונים. בקוד השתמשנו בנתון כי בשתי מאגרי הנתונים יש לנו את השמות של האנשים בעלי הזהויות, יצרנו רשימה של שמות של אנשים שנמצאים במאגר הנתונים של האודיו, לאחר מכן יצרנו רשימה של שמות של אנשים שנמצאים במאגר הנתונים של התמונות פרצוף, יצרנו רשימה שמהווה את החיתוך של שתי הרשימות הקודמות כך שקיבלנו רשימה של שמות של זהויות שנמצאת בשתי מאגרי המידע. לבסוף, עברנו על שתי המאגרים ובדקנו אם השם של הזהות הנוכחית נמצאת ברשימה של החיתוך, אם כן נכניס את רשימת המיקומים של כל הקבצים ששייכים לזהות זו למאגר נתונים מעודכן, כך שלבסוף קיבלנו נתונים רק עבור זהויות של אנשים שנמצאות בשתי מאגרי המידע ויש לנו עבורם גם הקלטות וגם תמונת פרצוף.

20 אלגוריתם האימון

כעת, נסביר אודות תהליך האימון של הרשתות והנוסחאות המשמשות אותנו. אלגוריתם אימון של הרשתות Generator, Discriminator, Classifier מתואר מטה באיור 12.

Algorithm 1 The training algorithm of the proposed framework

Input: A set of voice recordings with identity label $(\mathcal{V}, \mathcal{Y}^v)$. A set of labeled face images with identity label $(\mathcal{F}, \mathcal{Y}^f)$. A voice embedding network $F_e(v;\theta_e)$ trained on $\mathcal{V}$ with speaker recognition task. θ_e is fixed during the training. Randomly initialized θ_g, θ_d, and θ_c

Output: The parameters θ_g.

1: **while** not converge **do**

2: Randomly sample a minibatch of n voice recordings $\{v_1, v_2, ..., v_n\}$ from $\mathcal{V}$

3: Randomly sample a minibatch of m face images $\{f_1, f_2, ..., f_m\}$ from $\mathcal{F}$

4: Update the discriminator $F_d(f;\theta_d)$ by ascending the gradient

$$\nabla_{\theta_d}\big(\sum_{i=1}^{n}\log(1-F_d(\hat{f}_i))+\sum_{i=1}^{m}\log F_d(f_i)\big)$$

5: Update the classifier $F_c(f;\theta_c)$ by ascending the gradient ($a[i]$ indicates the i-th element of vector a)

$$\nabla_{\theta_c}\big(\sum_{i=1}^{m}\log F_c(f_i)[y_i^f]\big)$$

6: Update the generator $F_g(f;\theta_c)$ by ascending the gradient

$$\nabla_{\theta_g}\big(\sum_{i=1}^{n}\log F_c(F_g(F_e(v_i)))[y_i^v]+\sum_{i=1}^{m}\log F_d(F_g(F_e(v_i)))\big)$$

7: **end while**

איור 12: אלגוריתם האימון של המערכת המוצעת

כפי שמתואר באלגוריתם, הקלט הוא קבוצה (batch) של קבצי שמע ותמונות פרצוף המתאימות להן. בנוסף, הקלט מכיל רשת נוירונים אשר אומנה מראש על בעיית זיהוי דוברים על פי אותם קבצי אודיו (רשת ה- Voice Embedding).
נסביר אודות נוסחאות האלגוריתם וכיצד אימון לפיהן יניב את המטרה.
נוסחה ראשונה, המוצגת בשורה מספר 4. בנוסחה זו יש לנו שני גורמים. הגורם הראשון הוא תמונת פרצוף אמיתית, והשני הוא תמונת פרצוף מזויפת. כידוע ה- Discriminator מחזיר הסתברות בין 0 ל- 1 לסיווג תמונת פרצוף כאמיתית או מזויפת.
לכן ה- Discriminator רוצה למקסם את ההסתברות לזיהוי התמונה האמיתית כאמיתית (log הינה פונקציה מונוטונית עולה, ולכן אם נמקסם את log הביטוי נמקסם את גם את

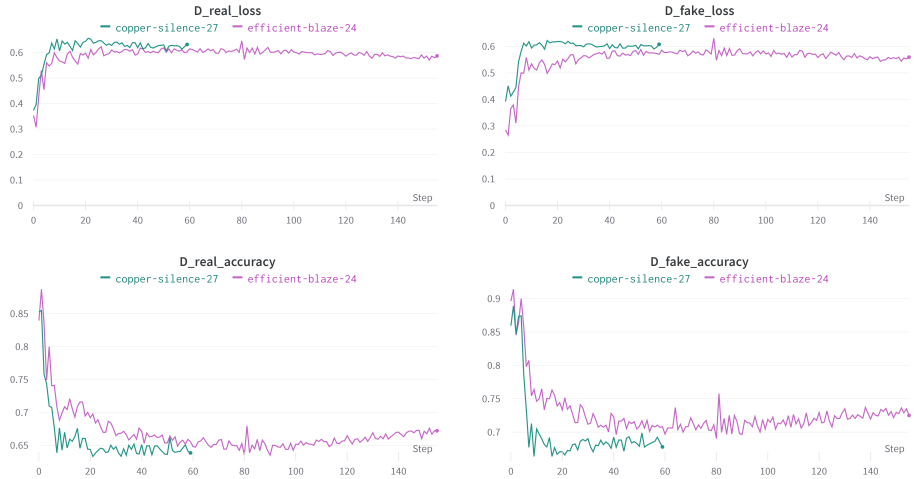
הביטוי עצמו). בנוסף, עבור תמונה מזויפת ה- Discriminator שואף להקטין את ההסתברות לזיהוי התמונה כאמיתית. באופן שקול, מזעור הביטוי של 1 פחות הסתברות זו. כך נקבל כי ההסתברות לזיהוי תמונה מזויפת תקטן.
נוסחה שנייה, המוצגת בשורה מספר 5. רשת Classifier מחזירה וקטור הסתברויות בו כל איבר מייצג את ההסתברות עבור שייכות ההקלטה לדובר מסוים. אנו רוצים במהלך האימון למקסם את ההסתברות לזיהוי נכון של הדובר האמיתי. כלומר, נרצה כי האיבר המתאים לדובר בוקטור ההסתברויות המתקבל על ידי ה- Classifier יהיה מקסימלי. שוב, השימוש בפונקציית log, הינו כי היא מונוטונית עולה.
לכן, נגדיל את הסתברות זו על ידי נתינת משקל גדול יותר לדובר האמיתי. כך נדייק את הפרמטרים הנלמדים כל פעם בהתאם לדובר האמיתי והחיזוי שלנו.
נוסחה שלישית, המוצגת בשורה מספר 6. בייצור התמונה לפי הקלטה שקיבלנו יש שתי מטרות. המטרה הראשונה היא שהתמונה תיראה אמיתית כמה שיותר, וגם שהתמונה תייצג את זהות הדובר ותהיה קשורה אליו, שכן אנו רוצים ליצור תמונות בעלות קשר להקלטה של הדובר. בנוסחה זו יש לנו שני גורמים. מטרת הגורם הימני היא להעלות את ההסתברות לייצור תמונה שתהיה דומה לתמונות האמיתיות כך שנצליח "לרמות" את ה- Discriminator בצורה יותר אמינה. מטרת הגורם השמאלי היא ליצור תמונה שתזוהה על ידי Classifier כתמונה בה הזהות של הדובר מתאימה לזהות האמיתית שיש לנו מהתיוג.
לאחר חישוב הנוסחה נעדכן את רשת ה- Generator כך שנוסיף את התוצאה שהתקבלה לפרמטרים של הרשת.
נחזור ונבצע שוב ושוב את הנוסחאות הללו עד להתכנסות, כלומר עד שהפרמטרים אשר אותם אנו מעדכנים כבר בקושי ישתנו ויגיעו לטווח ערכים מאוד מצומצם. כשנגיע לשם, נדע שקיבלנו דיוק יחסית גבוה ונעצור.

חלק V
תוצאות

בחלק זה נציג את תוצאות המודל. נעשה זאת על ידי הצגת גרפים של פונקציית ה- Loss עבור רשתות הנוירונים במערכת אשר מעידים על אימון נכון ולמידה של המודל. כמו כן, נראה את פלט רשת ה- Generator, תמונות פרצוף מזויפות ונראה כי מתקבל שיפור שלהן במהלך האימון.

21 Discriminator Loss And Accuracy

האיור מטה מתאר את פונקציית ה- Loss וה- Accuracy של רשת ה- Discriminator כתלות במספר ה- epoch.



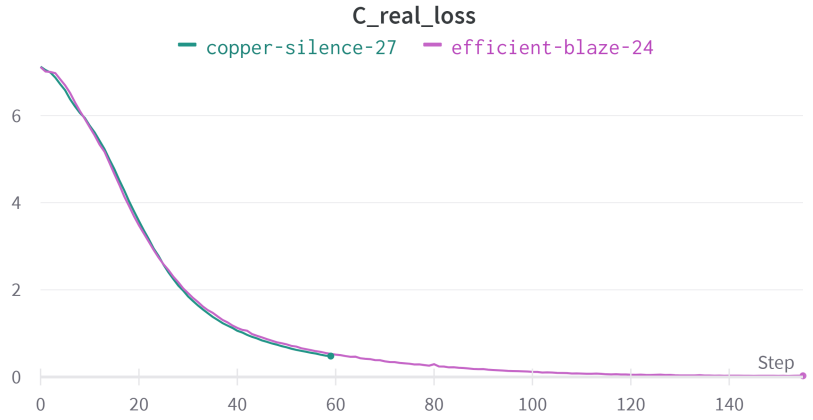
איור 13: פונקציות ה- Loss ו- Accuracy של רשת ה- Discriminator

קו ירוק וקו סגול מתארים את ערך ה- Loss ו- Accuracy בשתי ריצות שונות של אימון המודל. כפי שניתן לראות, ריצת הקו הסגול נמשכת על פני מספר רב יותר של epochs. ביתר פירוט, 160 אפוקים לעומת 60. מהגרפים ניתן להסיק שריצה של 60 אפוקים (קו ירוק) מספיקה לקבלת אופטימליות טובה של המודל. אחריה זה מטריקות המודל ד"י מתייצבות. נשים לב כי לרשת ישנה פונקציית Loss עבור תמונות אמיתיות ועבור תמונות מזויפות. דבר זה נכון מאחר ובכל batch ה- Discriminator מקבל תמונות אמיתיות וגם תמונות מזויפות אשר נוצרו על ידי ה- Generator. באופן זה, ניתן לחלץ פונקציית הפסד ודיוק עבור תמונות אמיתיות ועבור תמונות מזויפות.
כזכור ה- Discriminator מטרתו היא להגיד האם התמונה היא אמיתית או מזויפת והוא נותן פלט הסתברותי. במהלך האימון ה- Generator משתפר ומוציא תמונות פרצוף אמינות יותר כלומר, ההבחנה של ה- Discriminator בין תמונות אמיתיות למזויפות נהפכת קשה יותר. לכן, ה- Loss של ה- Discriminator עולה בהתקדמות תהליך האימון והדיוק יורד.

כידוע, כאשר המודל מאפטם את עצמו ה- Discriminator נותן פלט רנדומי. במקרה זה, ה- Loss שלו הוא המקסימלי וה- Accuracy שלו מינימלי.

22 Classifier Loss

האיור מטה מתאר את פונקציית ה- Loss של רשת ה- Classifier כתלות במספר ה- epoch.

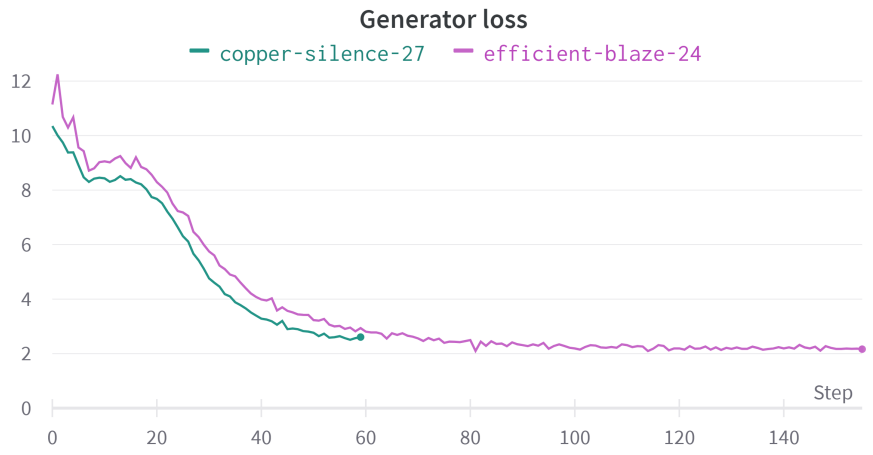


איור 14: פונקצית Loss של רשת ה- Classifier

בדומה לאיור 13, מתוארות אותן שתי ריצות שונות של אימון המודל. הפעם, אימון קצר של המודל אינו מספיק למזעור מקסימלי של פונקציית ה- Loss ונדרשים אפוקים נוספים.
עבור ה- Classifer יידרש תיוג תמונת הפרצוף כלומר, שיוך לדובר בדאטאסט וזאת כדי לבדוק את נכונות החיזוי שלו.
לכן, נבדוק את נכונות ה- Classifier על פי הדאטא האמיתי דהיינו, התמונות שיש להן תיוג.
כפי שניתן לראות אכן בהתקדמות מספר ה- epoch יש ירידה של ה- Loss דהיינו, החיזוי של אישות הדובר לתמונה המתקבלת משתפר, דיוק ה- Classifier עולה.

23 Generator Loss And Output

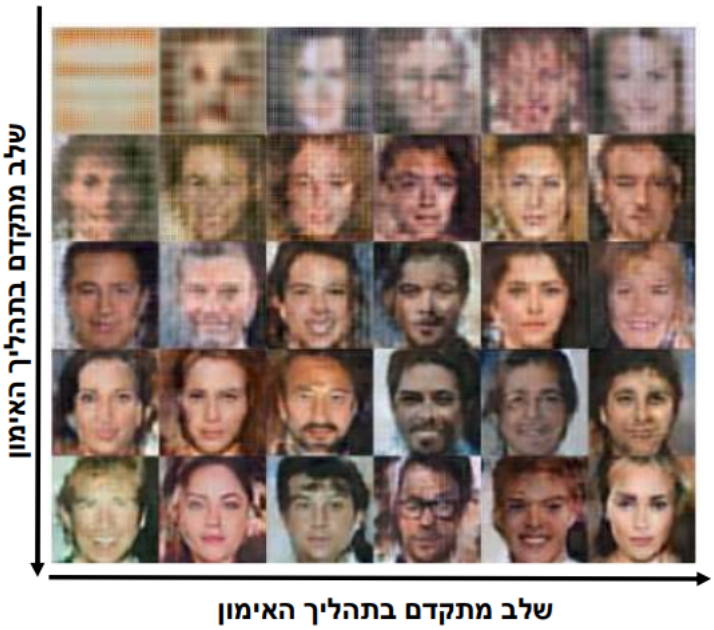
האיור מטה מתאר את פונקציית ה- Loss של רשת ה- Generator כתלות במספר ה- epoch.



איור 15: פונקצית Loss של רשת ה- Generator

גם כאן, מתוארות אותן שתי הריצות של אימון המודל. אימון קצר של המודל אינו מספיק למזעור מקסימלי של פונקציית ה- Loss ונדרשים אפוקים נוספים.
נראה כי במהלך האימון ה- Generator משתפר, הוא מוציא דוגמאות טובות יותר הקשות יותר להבחנה על ידי ה- Discriminator. לכן, ה- Loss שלו יורד.
נשים לב, כי קיבלנו דבר הגיוני. כזכור, הרשתות Generator ו- Discriminator מתחרות זו בזו.
ניצחון של רשת ה- Generator, מזעור פונקציית ה- Loss שלה. הוא הפסד של רשת ה- Discriminator, עליית פונקציית ה- Loss שלה. דהיינו, הגיוני כי פונקציות ה- Loss נגדיות זו לזו, האחת עולה והשנייה יורדת.

האיור מטה מתאר תמונות מזויפות אשר נוצרו על ידי ה- Generator כתלות במספר ה- epoch.



איור 16: תמונות המתקבלות על ידי רשת ה- Generator כתלות במספר ה- epoch.
תמונה מופיעה כל 2 אפוקים. הסדר הוא משמאל לימין למעלה למטה.

ניתן לשים לב כי במהלך התקדמות האימון ה- Generator מוציא תמונות שהן יותר הגיוניות, אמיתיות. הן אכן דומות לתמונות פרצוף.
כמו כן, נשים לב כי בתחילת האימון ה- Generator מפיק דוגמאות דאטא גרועות, זבל רנדומי. לאט לאט עם התקדמות מספר ה- epoch ניכר שיפור בתמונות הסינטטיות הנוצרות על ידו.

לסיום, נציג את פלט ה- Generator עבור הקלטות אודיו שלנו אשר הוכנסו כקלט למערכת.

איור 17: תמונות המתקבלות על ידי רשת ה- Generator כתוצאה מהקלטותינו.
מצד ימין תמונות פלט הרשת מצד שמאל תמונות אמיתיות שלנו. למעלה עידו, למטה עילי.

ניתן לשים לב כי התמונות אשר יוצרו על ידי ה- Generator זהות לתמונות שלנו עד כדי מאפיינים מרכזיים כגון: מין, צבע עור ואפילו תווי פנים. כצפוי הפלט אינו מושלם אך, תוצאה זו ד"י טובה.
כמו כן, תמונות הפרצוף אשר נוצרו על ידי הקלטות שמע שונות שלנו היו יחסית זהות. דבר אשר מעיד על קונסיסטנטיות של המודל.

חלק VI
מסקנות והצעות לשיפור

בחלק זה נפרט על מסקנותינו מפרויקט זה, תובנות, כלים שלמדנו ועוד. כמו כן, נציע הצעות לשיפור ולכיוון מחקר עתידי מתוך החומר המבוסס בעבודה זו.

24 מסקנות

עוד לפני שהתחלנו לבצע את המשימה, לייצר תמונת פנים של אדם על פי קולו, היה ברור לנו שיש מאפיינים בפנים שאינם מושפעים מהקול. למשל, לצבע השיער או לסוג השיער (חלק, מתולתל) אין קשר לקול של האדם. ציפינו בעיקר לקבל מתוך התוצאות מאפיינים בסיסיים אשר דומים לפרצופו האמיתי של הדובר למשל, מבנה פנים דומה, מין של הדובר. מאפיינים אלו יצאו מדוייקים דבר המעיד על כך שהרשת שבנינו למדה ליצור מאפיינים נכונים על סמך המידע שקיבלה. בגלל שפנים של אדם מושפעים מהרבה גורמים (גנים, מוצא, צלקות, השתלת סיליקון, פנים עם כווית ועוד) יש הרבה צורות של פנים, ולכן המידע שאיתו אימנו בוודאי לא מספיק נרחב בשביל לקבל תוצאות בדיוק גבוה. כמו כן, דבר נוסף אשר הוריד מדיוק התוצאות הוא כמות השכבות שעליהן אימנו. לרוב, כאשר מאמנים על יותר שכבות מגיעים לקשרים סטטיסטים יותר מדוייקים כלומר, לדיוק גבוה יותר של פרמטרי הרשת. עם זאת, נרצה מודל שיוכל להתאמן על הדאטא בזמן סביר (בעקבות מגבלות זמן, חישוב וזיכרון) ועל כן נצטרך לצמצם באיכות המודל כדי לעמוד במגבלות החישוביות. כשבחנו את תוצאות האימון של הרשת שמייצרת תמונות, הסתכלנו על שתי גרפים מרכזיים שהם הדיוק של התמונות שיוצרו ביחס לזיהוי זהות הדובר וביחס לשאלה האם התמונה שהרשת יצרה נראית אמיתית. הדיוק שיצא ביחס לזהות הדובר יצא גבוה יותר מהדיוק ביחס לתמונה אמיתית או מזויפת. דבר נוסף שבחנו הוא האם המודל יהיה קונסיסטנטי, כלומר אם נכניס לרשת כמה הקלטות של אותו דובר עם צלילים שונים האם הוא יפיק תמונות דומות, וראינו שהוא אכן עושה את זה. בנוסף, הכנסנו לו הקלטות של חיות אחרות, למשל צלילים של כלבים וחתולים, והוא ייצר פנים שנראות לא טובות ולא אמיתיות, ואכן כך ציפינו שיהיה. לעומת זאת, המודל כשל למשל בזיהוי גוון העור של הדובר, למשל שהכנסנו לו הקלטה של נשיא ארה"ב ברק אובמה קיבלנו פנים של אדם בעל צבע עור לבן. להערכתנו נובע מכך שאין מספיק גיוון בדאטאסט והמודל לא מאומן מספיק בשביל להבחין במאפיין הזה (גם לא ברור לנו כל כך כיצד הוא משפיע תמיד). למעשה בשביל לקבל מודל יותר מדוייק היינו בוחנים את הדיוק באופן ספציפי על רשימת מאפיינים ויזואליים בפרצוף של אנשים, וזאת משימה הרבה יותר מסובכת, עקב עניינים שכבר הוזכרו כמו דיוק ומגבלות חישוב. נשאלת השאלה גם מה יקרה אם ננסה לבחון את המודל על הקלטה של אדם מגמגם והאם זה משפיע על התוצאות, לצערנו לא העמקנו במחקר על העניין הזה אבל זה נקודה שניתן לבדוק גם. שאלה ששאלו אותנו הרבה במהלך הכנת הפרוייקט היא האם אנחנו מאמינים שיהיה ניתן בעתיד לקבל תמונת פנים מדויקת של אדם על סמך הקול שלו, שזו משימה חשובה למשל, כדי לפענח פשעים ובשביל משימות נוספות, והתשובה שלנו היא שאם נתמקד במאפיינים שאנחנו כן יכולים לחלץ מהקול של אדם ונגיע לרמת דיוק גבוהה מאוד, הדבר יעזור מאוד גם בלי לדעת את המאפיינים החסרים, שכן אם נדע את מבנה הפנים של האדם כנראה נדע לזהות אותו מבין קבוצה מצומצמת של אנשים, ולכן אנחנו מניחים שזה יסייע בעתיד בתחום המשפטי וגם בתחומים נוספים.

25 הצעות לשיפור

תחילה, כפי שציינו במסקנות ניתן להוסיף שכבות עד לקבלת ארכיטקטורה אופטימלית של הרשתות. לדוגמה, הוספת שכבות קונבולוציה לרשת ה- Generator ייעל את אופן חילוץ הפיצ'רים מתוך וקטור הקידוד דבר זה יכול להניב תוצאות איכותיות יותר.
רשת ה- Voice Embedding אחראית על קידוד ההקלטה המתקבלת של הדובר. רעיון לכיון מחקר הוא לשפר רשת זו שתתמוך בהקלטות בעלות רעשי רקע יותר חזקים, דיבור לא רציף (דובר מגמגם). כמובן שזה דבר מאתגר מאחר וזאת רשת אשר מאומנת מראש על הדאטא ובמקרה זה, נשאף לעבוד עם רשת אשר ראתה כמה שיותר דוגמאות מתויגות ולמדה כמה שיותר מקרי קצה. כמו כן, רשת זו תוכל לתת משקל גדול יותר לצלילים מסוימים בעלי קורלציה גדולה יותר לפרצוף.
עבור ה- Discrminator ו- Classifier אשר מסווגות תמונות נוכל להעזר ב- ResNet, רשת נוירונים שאומנה מראש על יותר ממיליון תמונות ותוכל לשפר את דיוק רשתות אלו.
נוכל להוסיף רשתות Classifier נוספות הדומות לרשת Classifier שלנו שמטרתן לזהות מאפיינים נוספים של הדברים שנתונים לנו כגון: מין, מדינת מוצא ועוד. מאפיינים אלו כלולים בדאטאסט וכך נוכל להיעזר בהם לשיפור המודל.

פרויקט זה מוקדש למשפחותינו היקרות.

ברצוננו להודות בראש ובראשונה למנחה שלנו מר יוחאי ימיני
על עזרתו בכל צעד ושלב בדרך.

ברצוננו להודות לפרופ' שרון גנות ולסגל הנדסה
באוניברסיטת בר-אילן על האפשרות לעסוק, לחקור
ולהעמיק בנושאי מחקר אלו ולצבור ידע רב וחשוב.

ברצוננו להודות לחוקרים שחקרו נושא זה לפנינו ונתנו לנו
בסיס טוב לפרויקט הזה.