

Deep Learning Assignment 2

אלעד גוברמן 209320696 , ניב כהן 318507241

ניתוח דאטה

Dataset

LFW Dataset Statistics:

```
Total number of images: 13233
Total number of people: 5749
Average images per person: 2.30
Median images per person: 1.00
Minimum images for a person: 1
Maximum images for a person: 530
Number of people with only one image: 4069
```

במסגרת העבודה קיבלנו את מאגר התמונות LFW המשמש בתחום זיהוי פנים. לאחר ניתוח הדאטה גילינו כי המאגר מכיל 13,233 תמונות פנים של 5,749 אנשים שונים. בנוסף ראינו כי קיים חוסר איזון במאגר. 4,069 אנשים (כ-70% מהמאגר) מיוצגים ע"י תמונה בודדת ומצד השני ישנם אנשים בודדים שיש להם עד 530 תמונות במאגר.

ניתוח סט האימון - סט האימון מכיל 2,200 זוגות של תמונות לאימון המודל.

הזוגות מחולקים לשני סוגים: 50% מהזוגות חיוביים ו-50% שליליים (1,100 בכל קבוצה).

נגדיר זוג תמונות חיובי כזוג תמונות המייצגות אותו אדם, דוגמה לכך מסט האימון היא שתי תמונות שונות של

: Aaron_Peirsol



נגדיר זוג שלילי כזוג תמונות המייצגות אנשים שונים, דוגמה לכך היא תמונה של AJ_Cook ותמונה של

: Marsha_Thomason



ניתוח סט הבדיקה (Test) –

סט זה מכיל 1000 זוגות של תמונות. בדומה לסט האימון, קיימת חלוקה לזוגות חיוביים ושליליים.

בכל קבוצה נמצאים 50% מסט הבדיקה (500 זוגות של תמונות).

מתודולוגיה ותהליך האימון

פיתחנו שלוש ארכיטקטורות שונות של רשת סיאמית לזיהוי פנים: אחת המבוססת על המאמר המקורי ושתיים נוספות שפיתחנו. להלן תיאור מפורט של תהליך העבודה, החל מעיבוד הנתונים הראשוני ועד לאימון והערכת המודלים.

1. עיבוד וטרנספורמציה של הנתונים:

שינוי גודל התמונה (Resize) – עבור ארכיטקטורת Paper בלבד ביצענו שינוי בגודל התמונות ל-105X105 פיקסלים

בהתאם למתואר במאמר. למרות שהסיבה להקטנת התמונות אינה כתובה באופן מפורש במאמר, אנחנו משערים כי שינוי התמונה במאמר ל-105X105 התבצע משתי סיבות עיקריות :

- i. איזון בין שימור מידע לבין יעילות חישובית - הקטנת התמונות מקטינה משמעותית את כמות החישובים הנדרשת תוך שמירה על רמת פירוט מספקת לזיהוי תווי פנים.
- ii. התאמה למאגר הנתונים Omniglot - מאגר הנתונים Omniglot, שבו השתמשו המחקרים, מכיל תמונות בגווי אפור של תווים כתובים ידנית. הגודל המקורי של התמונות במאגר הוא 105X105, ולכן התאמה לגודל זה שומרת על הפרופורציות המקוריות של התמונות בלי לבצע עיבוד יתר או הפחתת פרטים, וכך שומרת על המידע הוויזואלי החשוב בתווים.

ב-2 הארכיטקטורות החדשות שפיתחנו, **בחרנו לשמר את הרזולוציה המקורית של התמונות** משתי סיבות עיקריות :

- i. ההנחה כי שמירה על התמונה המקורית תשמור על פרטים ויזואליים שעשויים להיות קריטיים לזיהוי דמיון או שוני בין פנים והצלחת המודל.
- ii. ניצול היכולות המשופרות של הארכיטקטורות החדשות, המסוגלות לעבד תמונות ברזולוציה גבוהה יותר בזכות שימוש בשכבת Adaptive Average Pooling ופילטרים קטנים יותר בשכבות הקונבולוציה.

נרמול הנתונים – בהתאם למתואר במאמר המקורי, תהליך נרמול הנתונים כלל המרת התמונות למבנה טנסורי וביצוע נורמליזציה. בפועל, עבור כל תמונה חישבנו את הממוצע וסטיית התקן של ערכי הפיקסלים, הפחתנו את הממוצע מכל פיקסל וחילקנו בסטיית התקן. תהליך זה מוביל לערכים המתפלגים סביב 0 עם סטיית תקן של 1. הערכים השליליים שיכולים להתקבל מאפשרים השפעה משמעותית יותר בפונקציית האקטיבציה ReLU אשר יכול לעודד feature selection ברשת, שכן הוא מאפשר לחלק מהפיצ'רים להתאפס כאשר אינם תורמים משמעותית לתהליך הזיהוי.

2. ארכיטקטורת המודל:

כאמור, השווינו בין שלוש ארכיטקטורות שונות של רשת סיאמית. בבניית הארכיטקטורות החדשות התמקדנו

Paper Architecture

Layer	Input Size	Kernel Size	Num Filters	Max Pooling Size	Activation Function	Num Parameters
Conv1	1x105x105	10x10	64	2x2	ReLU	6,464
Conv2	64x48x48	7x7	128	2x2	ReLU	401,536
Conv3	128x21x21	4x4	128	2x2	ReLU	262,272
Conv4	128x9x9	4x4	256	N/A	ReLU	524,544
FC1	9,216	N/A	4,096	N/A	Sigmoid	37,752,832
FC2	4,096	N/A	1	N/A	Sigmoid	4,097

New Architecture 1

Layer	Input Size	Kernel Size	Num Filters	Max Pooling Size	Activation Function	Num Parameters
Conv1	1x250x250	5x5	64	2x2	ReLU	1,664
Conv2	64x123x123	5x5	128	2x2	ReLU	204,928
Conv3	128x59x59	3x3	128	2x2	ReLU	147,584
Conv4	128x28x28	3x3	256	N/A	ReLU	295,168
AdaptiveAvgPool	256x28x28	N/A	N/A	N/A (output: 6x6)	N/A	0
FC1	9,216	N/A	2,048	N/A	ReLU	18,876,416
FC2	2,048	N/A	1	N/A	Sigmoid	2,049

New Architecture 2

Layer	Input Size	Kernel Size	Num Filters	Max Pooling Size	Activation Function	Num Parameters
Conv1	1x250x250	5x5	64	2x2	ReLU + BatchNorm	1,664 + 128
Conv2	64x123x123	5x5	128	2x2	ReLU + BatchNorm	204,928 + 256
Conv3	128x59x59	3x3	128	2x2	ReLU + BatchNorm	147,584 + 256
Conv4	128x28x28	3x3	256	N/A	ReLU	295,168
AdaptiveAvgPool	256x28x28	N/A	N/A	N/A (output: 6x6)	N/A	0
FC1	9,216	N/A	2,048	N/A	ReLU + BatchNorm	18,876,416 + 4,096
FC2	2,048	N/A	1	N/A	Sigmoid	2,049

iv. הוספת BatchNorm (רק ב-new_arch_2) – מאפשר קצב למידה גבוה תוך שמירה על יציבות המודל.

i. שימוש בפילטרים קטנים יותר

(3X3, 5X5) - מאפשר למידת מאפיינים מקומיים יותר.

ii. AdaptiveAvgPool - מאפשר

עבודה עם תמונות בגודל המקורי

(ובכללי עם רזולוציות גבוהות),

מגמיש את השימוש במודל (אינו

תלוי בגודל הקלט).

iii. הקטנת מספר הפרמטרים בשכבות

ה-FC (מ-4096 ל-2048) – הפחתה

משמעותית במס' הפרמטרים

במודל (הפחתה של כ-50%) שכן

מודל מסוג זה נוטה להתמודד טוב

יותר עם overfitting שיכול לנבוע

מעודף פרמטרים, במיוחד במאגרי

נתונים קטנים.

Feature	Paper Architecture	New Architecture 1	New Architecture
Input Size	105x105	250x250	250x250
Kernel Sizes	Large (10x10, 7x7, 4x4)	Small (5x5, 3x3)	Small (5x5, 3x3)
Pooling Method	Max Pooling	Adaptive Avg Pool	Adaptive Avg Pool
Hidden Layer Activation	ReLU + Sigmoid	ReLU	ReLU
FC Layer Size	4,096	2,048	2,048
Batch Normalization	No	No	Yes
Weight Initialization	Normal ($\mu=0, \sigma=0.01$)	Xavier Uniform	Xavier Uniform
Total Parameters	~39M	~19.5M	~19.5M

3. היפר-פרמטרים ואסטרטגיית אימון

על מנת לבדוק מה שילוב ההיפר-פרמטרים האופטימלי, הגדרנו מס' אופציות להיפר פרמטרים וע"י כך בחנו שילובים שלהם בניסויים שונים.

- i. batch size – נבדקו 3 גדלים שונים [32, 64, 128]. בחרנו מספר גדלים, שכל אחד משפיע על קצב ההתכנסות. נבדקו 3 גדלים שונים [128, 64, 32]. בחרנו מספר גדלים, שכל אחד משפיע על קצב ההתכנסות.
- Batch size קטן מכיל פחות דוגמאות ולכן עדכון המשקולות מתרחש בתדירות גבוהה יותר בכל Epoch. זה מאפשר למודל להגיב מהר יותר לשינויים בדאטה, מה שעשוי להוביל להתכנסות מהירה יותר (מספר גדול יותר של עדכונים לכל Epoch). יחד עם זאת, היענות מוגזמת עלולה לגרום לתנודתיות באימון ולא להתכנסות חלקה. ב Batch size גדול עדכון המשקולות מבוסס על מידע סטטיסטי מדויק יותר, מה שעשוי להוביל להתכנסות יציבה אך איטית יותר.
- ii. Epochs – בחרנו ב- 50. הבחירה במס' זה הייתה שרירותית אך מתוך מחשבה שצריך להיות גדול מספיק שתבצע למידה ובשאיפה תיעצר לפני כן (בזכות מנגנון Early Stopping) וגם קטן מספיק כדי שזמן הריצה יהיה סביר.
- iii. Learning Rate – נבדקו 3 ערכים שונים [1e – 3, 1e – 4, 1e – 5]. כך יכולנו לבצע השוואה ולראות עבור איזה ערך יתקבל המודל הטוב ביותר.
- iv. Optimizer – בחרנו באלגוריתם ADAM – נפוץ בתחום ומתאים למשימת זיהוי פנים.
- v. Early Stopping – 2 פרמטרים מרכזיים: **Patience** - מספר Epochs הכולל שבהן לא נצפה שיפור בביצועים (בחרנו ש- 5 יהיה המספר המקסימלי) ו- **Tolerance** - השינוי המינימלי ב-loss שנחשב כשיפור (בחרנו ב- 0.001). בדקנו כמה ערכים שונים ולבסוף בחרנו בערך זה שהוא קטן מספיק ו-patient גדול מספיק ע"מ שהאימון לא יסתיים מהר מדי. שילוב ערכים אלו הוביל לאיזון בין זמן אימון מספק לבין מניעת overfitting.
- vi. L2 Regularization – פרמטר רגולריזציה שמטרתו היא שמירה על משקלים קטנים באמצעות "הענשה" של משקלים גדולים וע"י כך נשמרת יציבות המודל והפחתה של overfitting. לאחר ניסוי וטעיה בחרנו ב- $\lambda = 10^{-5}$.

את הנתונים חילקנו באופן הבא :

- i. סט אימון : 80% מסט האימון המקורי (1,760 זוגות)
- ii. סט ולידציה : 20% מסט האימון המקורי (440 זוגות)
- iii. סט מבחן : נשאר כפי שהוא (1,000 זוגות)

אתחול המשקולות נעשה ב-2 אופנים שונים :

- i. אתחול התואם לנעשה במאמר (פונקציית initial_weights) – התפלגות נורמלית עם בחירת ממוצע וסטיות תקן שונים לשכבת הקונבולוציה ולשכבות הלינאריות.
- ii. אתחול Xavier (פונקציית initialize_weights_xavier) – מתבסס על הנוסחה

$$W \sim u\left(-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}}\right)$$

ומתאים במיוחד לעבודה עם ReLU. כמו כן, סוג אתחול זה מסייע במניעת בעיית vanishing/exploding gradients על ידי התאמת טווח המשקולות להתחלתי למספר הקלטים והפלטים בכל שכבה.

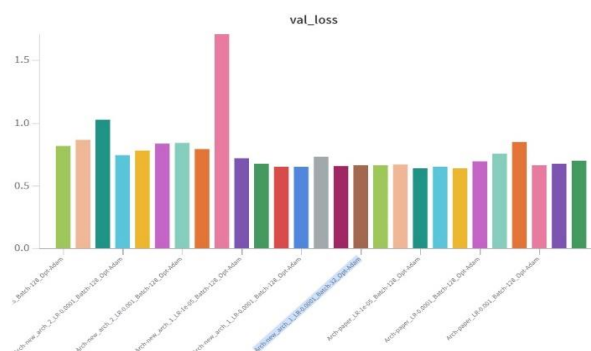
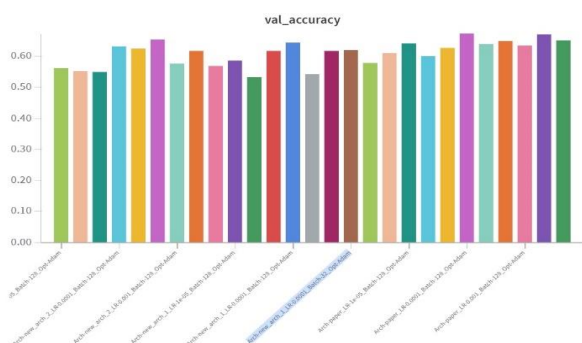
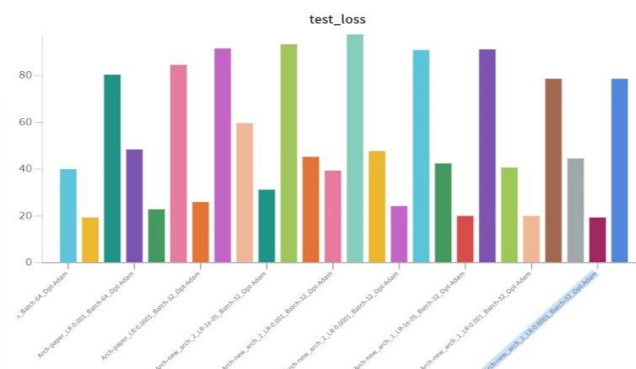
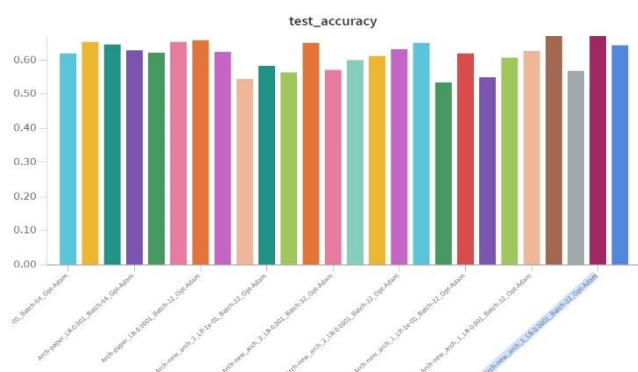
פונקציית הפסד היא Binary Cross-Entropy המודדת את ההבדל בין התפלגות הפלט של המודל להתפלגות האמיתית, כאשר יש רק שתי אפשרויות תוצאה (במקרה שלנו : האם זה אותו אדם או לא).

4. תהליך האימון –

במהלך המחקר בחנו 27 קונפיגורציות שונות של מודלים, שנוצרו משילוב של שלוש ארכיטקטורות עם ערכים שונים של learning rate וגדלי batch. תהליך האימון לכל מודל כלל : אימון על 1,760 זוגות תמונות תוך ניטור מתמיד של loss ו-accuracy באמצעות מערכת wandb, עד לעצירת האימון (על ידי Early Stopping או השלמת מספר Epochs המקסימלי), הערכה על סט ולידציה (440 זוגות) ולבסוף הערכה על סט המבחן (1,000 זוגות) הכוללת יצירת confusion matrix ושמירת דוגמאות לכל קטגוריות הסיווג. פונקציית grid_search בחנה את כל קומבינציות ההיפר-פרמטרים עבור כל ארכיטקטורה ובחרה את הקונפיגורציה המיטבית לפי accuracy (ובמקרה של שוויון, לפי loss). לבסוף, פונקציית get_best_architecture השוותה בין שלושת המודלים המובילים לפי ארכיטקטורה לבחירת המודל הסופי.

תוצאות ומסקנות

תוצאות כלליות –



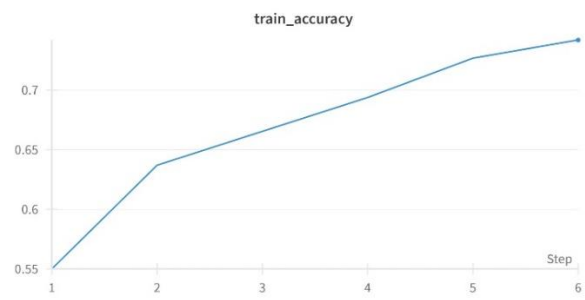
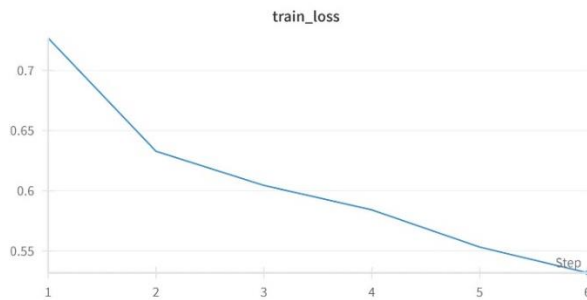
מתוצאות אלו ניתן לראות שמרבית המודלים השיגו accuracy של כ-60% הן בסט הוולידציה והן בסט המבחן. בולט במיוחד הדמיון בין מדדי הביצוע בוולידציה ובטסט, מה שמעיד על יציבות המודלים והיעדר overfitting משמעותי.

המודל הטוב יותר –

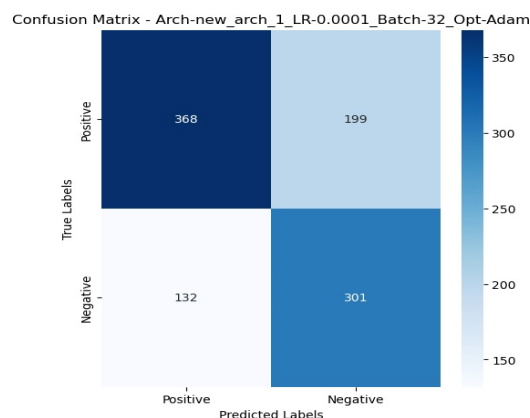
Config parameters: {} 4 keys architecture: "new_arch_1" batch_size: 32 learning_rate: 0.0001 optimizer: "Adam"	Summary metrics: {} 7 keys test_accuracy: 0.669 test_loss: 19.267980992794037 train_accuracy: 0.7420454545454546 train_loss: 0.5319192588329316 training_runtime: 60.53631591796875 val_accuracy: 0.6159090914509513 val_loss: 0.6554258953441273
---	---

לאחר סיום הרצת כל הניסויים קיבלנו את המודל הטוב ביותר. מודל זה הוא בעל הפרמטרים הבאים:
Architecture = "new_arch_1", batch size = 32, learning rate = 0.0001, Optimizer Adam

המודל הגיע לביצועים יציבים עם accuracy של 66.9% על סט המבחן ו-61.5% על סט הוולידציה בזמן ריצה של כדקה ולאחר 6 Epochs בלבד.



ניתן להסיק מגרפים אלו כי המודל למד בהדרגה לאורך כל תהליך האימון: ה- train accuracy עלה בעקביות מ-55% עד ל-74.2%, בעוד ה- train loss ירד באופן מונוטוני מ-0.7 ל-0.53, מה שמעיד על התכנסות טובה של המודל.



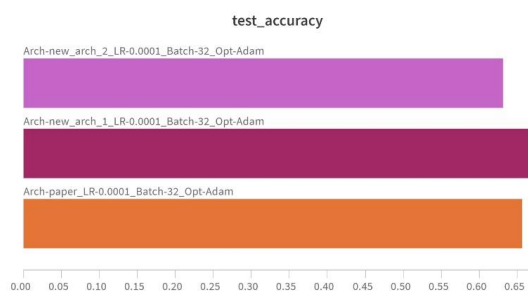
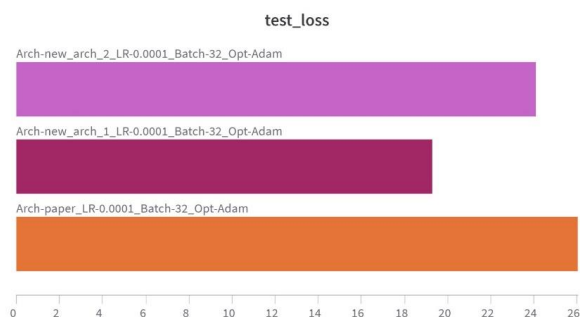
בנוסף מה- Confusion matrix ניתן לראות כי המודל הצליח לזהות באופן מאוזן יחסית הן זוגות זהים והן זוגות שונים, עם נטייה קלה להצלחה גבוהה יותר בזיהוי זוגות זהים (73.6% לעומת 60.2% בזיהוי זוגות שונים).

כחלק מתהליך האימון שמרנו חלק מסיווגי המודל (כל סיווג בנפרד) עבור כל ניבוי ותיוג. להלן דוגמא מסיווגי המודל הטוב ביותר:

	
<p>TP: דוגמא יפה של זיהוי מוצלח של המודל. בתמונה הימנית הבעת הפנים מכווצת יותר ויחד עם זאת המודל הצליח לזהות כי מדובר באותו אדם.</p>	<p>FN: דוגמא לחוסר הצלחה בזיהוי של המודל. אנו משערים שהמודל לא זיהה שמדובר באותו אדם בגלל הבדלים בין התמונות כגון: המשקפיים, צבע החולצה, בהירות הרקע וזווית הפנים.</p>
	
<p>TN: המודל הצליח לזהות כי מדובר באנשים שונים ואכן אין דמיון בין השניים.</p>	<p>FP: חוסר הצלחה של המודל. אנחנו משערים כי חוסר הצלחת המודל לזהות כי מדובר באנשים שונים היא בגלל תווי הפנים הדומות (אף, ועיניים).</p>

השוואות נוספות

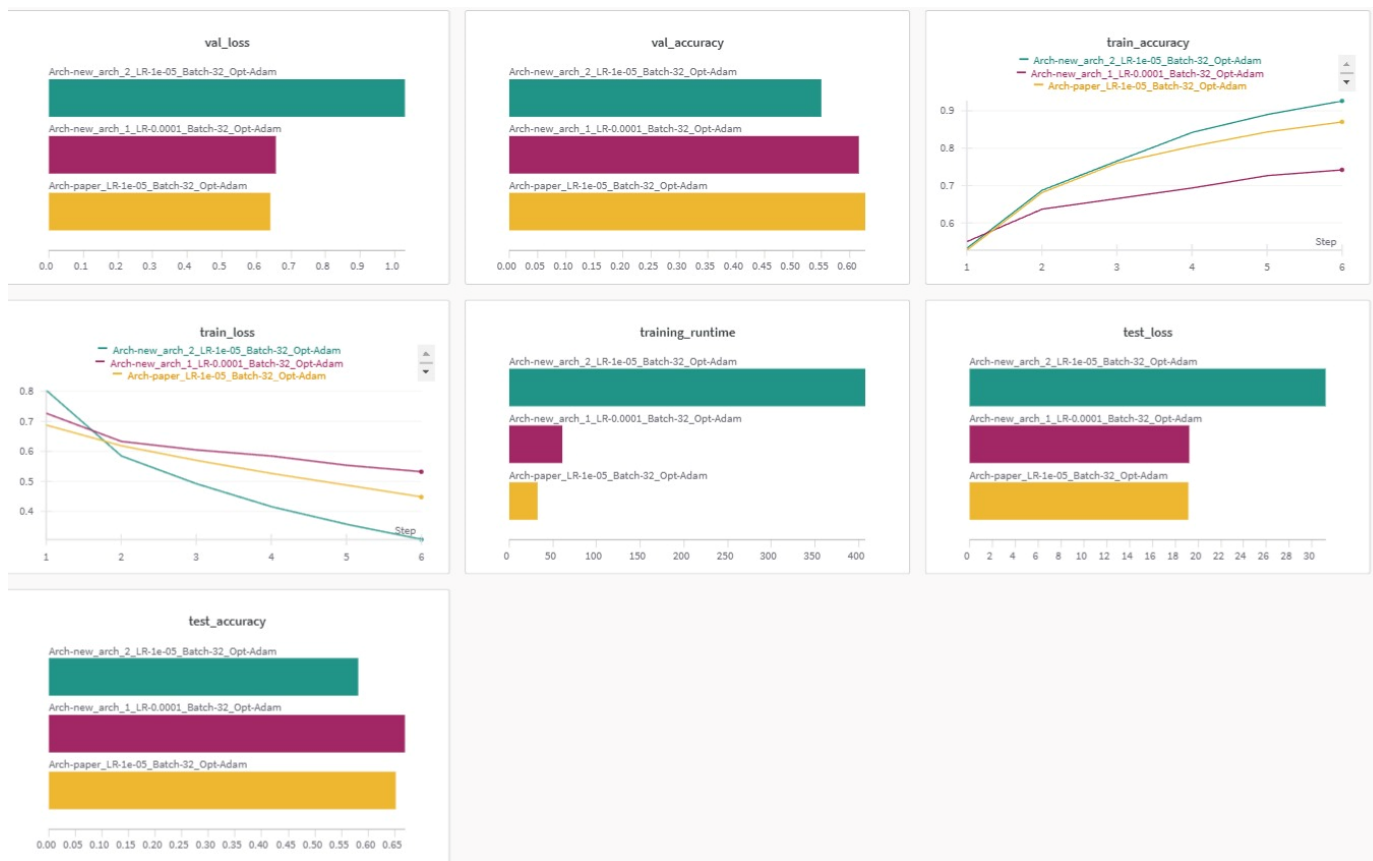
כדי להעמיק את הבנתנו בביצועי המודלים השונים, ערכנו השוואות נוספות בין המודלים השונים שהרצנו. רצינו לעשות השוואה מקיפה בין שלוש הארכיטקטורות תחת אותם היפר-פרמטרים שהניבו את התוצאות הטובות ביותר (Optimizer Adam, batch size=32, learning rate=0.0001). השוואה זו מאפשרת לנו לבחון את ההשפעה הישירה של השינויים הארכיטקטוניים על ביצועי המודל.



מהגרפים הכוללים את שלושת הארכיטקטורות הגענו למספר תובנות מעניינות:

- ברמת ה- test accuracy , כל הארכיטקטורות השיגו ביצועים דומים (סביב 65%). אך הארכיטקטורה המקורית מהמאמר הציגה loss גבוה יותר, מה שמעיד על ביטחון נמוך יותר בתחזיותיה ואילו הארכיטקטורה הראשונה שיצרנו הניבה loss נמוך יותר.
 - בגרפי האימון ניתן לראות הבדלים משמעותיים. הארכיטקטורה השנייה (new_arch_2) הפגינה למידה מהירה יותר והגיעה לדיוק גבוה יותר באימון (מעל 90%). המודל המקורי הראה התכנסות איטית יותר והגיע לדיוק נמוך יותר (סביב 75%). הירידה ב- train loss הייתה חדה יותר בארכיטקטורות החדשות, במיוחד ב- new_arch_2 .
- תוצאות אלו מדגישות את היתרון של הארכיטקטורות החדשות, ובמיוחד של שכבות ה-BatchNorm ב- new_arch_2 , בשיפור תהליך האימון, למרות שבסופו של דבר כל המודלים הגיעו לביצועים דומים על סט המבחן.

השוואה נוספת שעשינו היא השוואה בין המודל הטוב ביותר כפי שצינו בסעיף הקודם לבין המודלים הטובים ביותר של 2 הארכיטקטורות האחרות :



בהשוואה זו ניתן לראות דפוסים מעניינים. ראשית ההבדל המשמעותי ביותר הוא בזמני הריצה של כל אחד מהמודלים. הארכיטקטורה השנייה (new_arch_2) הציגה זמן ריצה ארוך משמעותית (כ-400 שניות, כמעט 7 דקות), כנראה בשל שכבות ה-BatchNorm הנוספות. אבל הגיעה להתכנסות הטובה ביותר ב-train loss. מסקנה נוספת שראינו כאן היא שהארכיטקטורה המקורית מהמאמר (paper), למרות פשטותה היחסית, השיגה ביצועי accuracy דומים בזמן ריצה קצר יותר, כלומר תהליך הלמידה היה מהיר יותר. תוצאות אלו מרמזות לנו שהמורכבות הנוספת של הארכיטקטורות החדשות שהוספנו לא תרמה בהכרח לשיפור בביצועים, אך כן השפיעה על יציבות האימון ועל זמני הריצה.

טבלה מסכמת של כל המודלים :

Name	architecture	batch size	learning rate	optimizer	test accuracy	test loss	train accuracy	train loss	training runtime	val accuracy	val loss
Arch-new_arch_1_LR-0.0001_Batch-32_Opt-Adam	new_arch_1	32	0.0001	Adam	0.669	19.26798099	0.742045455	0.531919259	60.53631592	0.615909091	0.655425895
Arch-new_arch_1_LR-0.001_Batch-128_Opt-Adam	new_arch_1	128	0.001	Adam	0.668	78.43512452	0.738068184	0.548046668	503.8577363	0.620454547	0.66572594
Arch-paper_LR-0.0001_Batch-32_Opt-Adam	paper	32	0.0001	Adam	0.657	26.03565735	0.977272727	0.105741762	32.69889855	0.647727274	0.84841593
Arch-paper_LR-0.001_Batch-128_Opt-Adam	paper	128	0.001	Adam	0.653	84.59523869	0.749431819	0.523779883	26.89002681	0.634090916	0.662886862
Arch-paper_LR-1e-05_Batch-32_Opt-Adam	paper	32	0.00001	Adam	0.653	19.17973974	0.870454545	0.447779532	32.50385356	0.627272729	0.640042921
Arch-new_arch_2_LR-0.001_Batch-64_Opt-Adam	new_arch_2	64	0.001	Adam	0.649	45.19090819	0.873295455	0.337270921	397.3550456	0.615909097	0.795423684
Arch-new_arch_2_LR-0.0001_Batch-128_Opt-Adam	new_arch_2	128	0.0001	Adam	0.649	90.95959419	0.998	0.059655981	707.571981	0.631818188	0.74579619
Arch-paper_LR-1e-05_Batch-128_Opt-Adam	paper	128	0.00001	Adam	0.644	80.20277256	0.794886364	0.545823854	27.15910125	0.640909092	0.64080265
Arch-new_arch_1_LR-0.0001_Batch-128_Opt-Adam	new_arch_1	128	0.0001	Adam	0.643	78.67088246	0.740909092	0.556071037	372.6893125	0.643181819	0.653921263
Arch-new_arch_2_LR-0.0001_Batch-32_Opt-Adam	new_arch_2	32	0.0001	Adam	0.631	24.10780713	0.998295455	0.055571993	412.0759239	0.654545457	0.838928185
Arch-paper_LR-0.001_Batch-64_Opt-Adam	paper	64	0.001	Adam	0.627	48.27261223	0.792613636	0.44902472	30.11134768	0.670454549	0.674850295
Arch-new_arch_1_LR-0.001_Batch-32_Opt-Adam	new_arch_1	32	0.001	Adam	0.626	19.78794074	0.647727273	0.620436923	120.5758259	0.609090911	0.672633277
Arch-paper_LR-0.0001_Batch-128_Opt-Adam	paper	128	0.0001	Adam	0.623	91.48381102	0.955113639	0.207760542	27.23304915	0.672727278	0.693812705
Arch-paper_LR-0.001_Batch-32_Opt-Adam	paper	32	0.001	Adam	0.62	22.90042803	0.803409091	0.429449918	33.28786683	0.650000002	0.701415264
Arch-paper_LR-1e-05_Batch-64_Opt-Adam	paper	64	0.00001	Adam	0.619	39.85063276	0.841477273	0.50670936	29.80279613	0.600000005	0.653405187
Arch-new_arch_1_LR-1e-05_Batch-32_Opt-Adam	new_arch_1	32	0.00001	Adam	0.618	20.07853252	0.669886364	0.623277438	60.54540896	0.61818182	0.652286677
Arch-new_arch_2_LR-0.0001_Batch-64_Opt-Adam	new_arch_2	64	0.0001	Adam	0.61	47.70784375	0.999431818	0.058973524	492.2639618	0.625000005	0.780822707
Arch-new_arch_1_LR-0.001_Batch-64_Opt-Adam	new_arch_1	64	0.001	Adam	0.607	40.64744112	0.601136364	0.646298368	214.7648389	0.577272731	0.663347154
Arch-new_arch_2_LR-0.001_Batch-128_Opt-Adam	new_arch_2	128	0.001	Adam	0.599	97.46980512	0.931250002	0.233050207	689.0365632	0.575000005	0.842842422
Arch-new_arch_2_LR-1e-05_Batch-32_Opt-Adam	new_arch_2	32	0.00001	Adam	0.582	31.1572901	0.926704545	0.30605401	407.1104755	0.550000002	1.028481258
Arch-new_arch_1_LR-0.001_Batch-32_Opt-Adam	new_arch_1	32	0.001	Adam	0.57	39.28814435	0.814204545	0.419741606	70.79231501	0.56818182	1.712389517
Arch-new_arch_1_LR-0.0001_Batch-64_Opt-Adam	new_arch_1	64	0.0001	Adam	0.569	44.32251069	0.596022727	0.673832489	214.4989548	0.540909094	0.729993344
Arch-new_arch_2_LR-1e-05_Batch-128_Opt-Adam	new_arch_2	128	0.00001	Adam	0.564	93.38947421	0.907386366	0.359554333	688.3514762	0.561363642	0.818056961
Arch-new_arch_1_LR-1e-05_Batch-128_Opt-Adam	new_arch_1	128	0.00001	Adam	0.549	91.01983851	0.55	0.74525042	374.7493019	0.586363637	0.719301005
Arch-new_arch_2_LR-1e-05_Batch-64_Opt-Adam	new_arch_2	64	0.00001	Adam	0.543	59.67205936	0.928409091	0.308246567	492.2617249	0.552272731	0.865907811
Arch-new_arch_1_LR-1e-05_Batch-64_Opt-Adam	new_arch_1	64	0.00001	Adam	0.533	42.34341255	0.5375	0.674080998	214.5475352	0.531818185	0.679265567

מתוך השוואת כלל המודלים ניתן לראות כי טווח ה- test accuracy נע בין 53% ל-67%, כאשר המודל המוביל new_arch_1 (עם batch size=32 ו- learning rate=0.0001) השיג 66.9%. מעניין לראות שהמודלים עם batch size קטן (32) נטו להשיג תוצאות טובות יותר, ושה- learning rate של 0.0001 היה אופטימלי ברוב המקרים. בנוסף, ניכר כי זמני הריצה של הארכיטקטורה השנייה היו ארוכים משמעותית מהאחרים, אך לא תרמו לשיפור בביצועים.

סיכום

בעבודה זו בחנו את משימת זיהוי הפנים באמצעות רשת סיאמית בגישת one-shot learning, תוך השוואה בין הארכיטקטורה המקורית מהמאמר לבין שתי ארכיטקטורות חדשות שפיתחנו. בחנו 27 קונפיגורציות שונות של מודלים, שנוצרו משילוב של שלוש ארכיטקטורות עם ערכים שונים של היפר-פרמטרים. המודל המוביל, המבוסס על הארכיטקטורה החדשה הראשונה, השיג accuracy של 66.9% על סט המבחן. למרות שהארכיטקטורות החדשות הציגו יתרונות בתהליך האימון ובהתכנסות, השיפור בביצועי המבחן היה מינורי (שיפור של כ-1.2%). ביחס לארכיטקטורה המקורית, במחיר של זמני אימון ארוכים יותר ומורכבות מודל גבוהה יותר.

התוצאות שקיבלנו מעלות שתי תובנות מרכזיות: ראשית, האתגר המשמעותי בזיהוי פנים בגישת one-shot learning, המצריך גישות חדשניות מעבר לשיפורים ארכיטקטוניים, ושנית, החשיבות של בחינת היחס בין תועלת לעלות בהוספת שכבות ומורכבות למודל. עם זאת, השיפורים שהצענו בארכיטקטורה, במיוחד בהיבטי יציבות האימון והתכנסות, מספקים תובנות חשובות למחקר עתידי בתחום.