



## למידה ייצוג וראיה ממוחשבת – מטלה 2



### מגישים:

אורן גולדפישר 307967612

חן גרינברג 204640262

### תאריך:

20/02/2021

תוכן ענייני

3	פרמטרים
4	שיטת כיוון הפרמטרים
5	כיוון הפרמטרים
8	המודל הנבחר וערכי הפרמטרים הנבחרים
9	שיפור המודל
11	שגיאות המודל

## פרמטרים

1. Width, Height – אלו הם גדלים קבועים, בהתאם לדרישות הרשת המקורית בה בחרנו להשתמש.
2. Optimization Method – השיטה בה המודל יילמד, כלומר, דרך הפעולה בה הוא יאמן את המשקולות של הרשת בשלב הTrain.
3. Decay – בערך זה ייתן קנס על המשקולות, זאת על מנת לחייב משקלים קטנים בפונקציית הLoss, כדי לצמצם את הסיכויים ל-over-fitting.
4. Learning Rate – ערך המייצג את המשקל שנקדיש לכל Sample בשלב האימון. כלומר, ככל שהערך גדול יותר, כך התנודות שנבצע במשקולות בעקבות Samples בשלב האימון, יהיו גדולים יותר.
5. Momentum – ערך זה מראה את קצב ההתכנסות של המודל, ככל שערך זה גבוה יותר, כך הוא "יתקדם" מהר יותר לנקודת ההתכנסות.
6. Epochs - כמות הפעמים בהם נעביר את כל Samples שלנו במודל בשלב האימון. ככל שכמות שהEpochs תהיה גדולה יותר, כך המודל יילמד טוב יותר את הדוגמאות מסט האימון שלנו, אבל, כמות גבוהה מדי של Epochs עלולה להוביל ל-over-fit, כיוון שהמודל לומד בצורה ספציפית מדי מודל עפ"י הדוגמאות שיש ברשותו.
7. Batch Size – קבוצת Samples שנכניס למודל בכל פעם שנלמד אותו, כלומר מה תהיה גודל הקבוצה (Batch) שנכניס למודל בכל איטרציה.
8. Early Stopping Criterion - רגולציה אשר מצמצמת את הסיכוי להגיע ל-over-fitting, הוא מופעל כאשר הloss לא משתפר לאחר מספר epochs.

### שיטת כיוון הפרמטרים

על מנת למצוא את המודל האופטימלי, ביצענו ולידציה רגילה. הוצאנו מתוך סט האימון שלנו 50 תמונות, אשר שימשו אותנו כסט ולידציה.

בכל פעם שרצינו למצוא פרמטר אופטימלי, הכנסנו את כל האופציות האפשריות, וראינו איזה מבין המודלים נותן לנו את התוצאה הטובה ביותר.

בסופו של דבר, הרצנו את המודל הנבחר על סט ה"טסט", 172 התמונות האחרונות בדאטה, והצגנו את התוצאות.



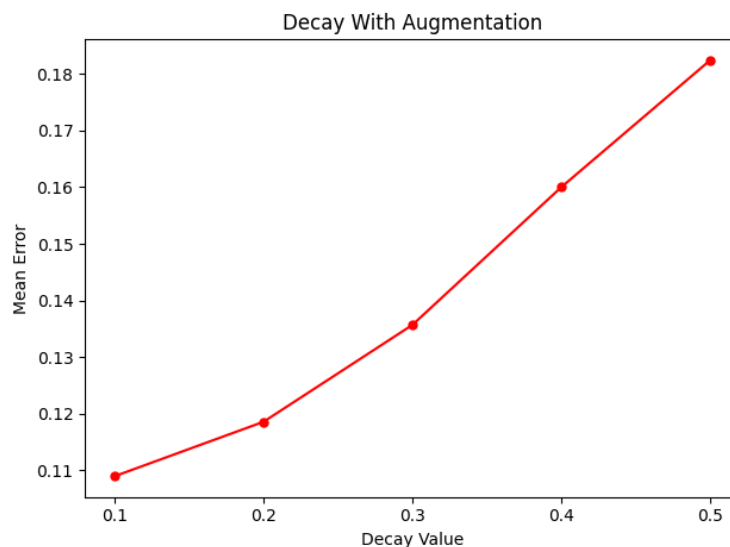
### כיוון הפרמטרים

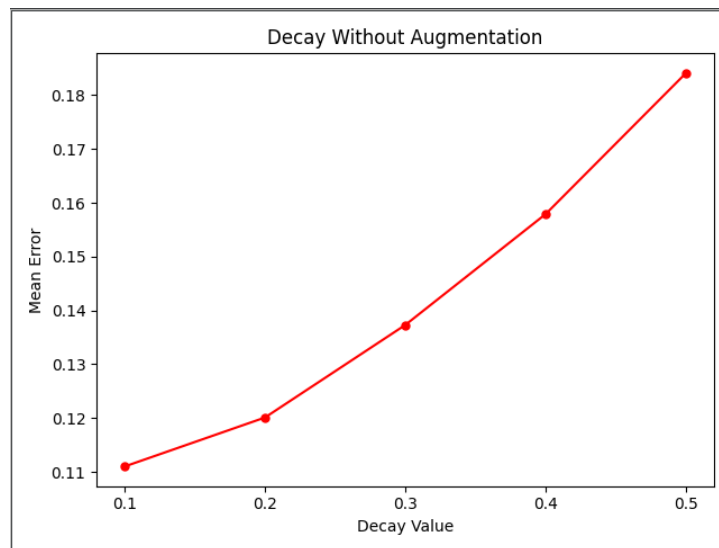
בתחילת העבודה על המודל, בחרנו לשפר את המודל בעזרת Data Augmentation, ואכן הגענו לאחוזי דיוק הגבוהים ביותר על סט הוולידציה. אך לאחר בדיקת זמן הריצה של המודל, עם הפרמטרים האופטימליים, זמן הריצה של המודל היה גבוה מהזמן שהוקצב לנו, ועל כן בחרנו בסופו של דבר לוותר על תהליך זה.

למרות זאת, מכיוון שהפרמטרים האופטימליים (פרט לכמות Epochs אשר האוגמנטציה באה על חשבונם), נציג את הגרפים עבור שני המודלים.

לאחר קריאה ובדיקה, בחרנו בפרמטרים שעפ"י דעתנו, כיוון שלהם, יוביל לשיפור המשמעותי ביותר במודל.

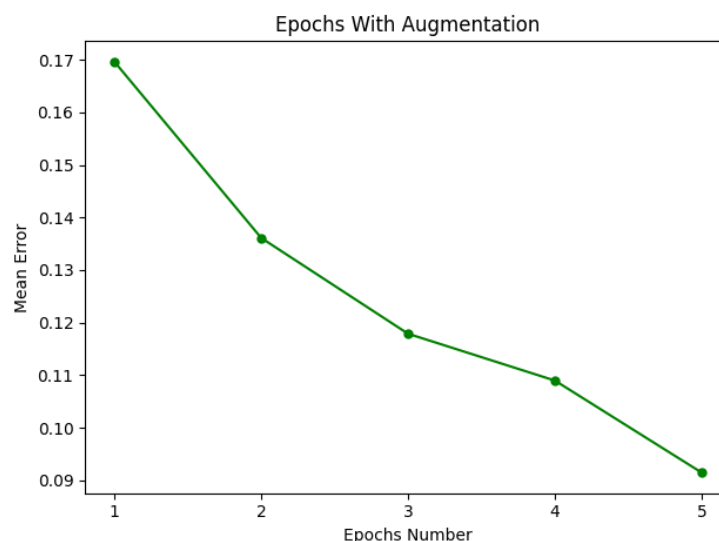
1. **Decay**: במקרה של היפר פרמטר זה, טווח הערכים בשני המקרים (עם אוגמנטציה ובלי) היה זהה, בין 0.1 ל 0.5, כאשר התקדמנו כל פעם בקפיצות של 0.1.

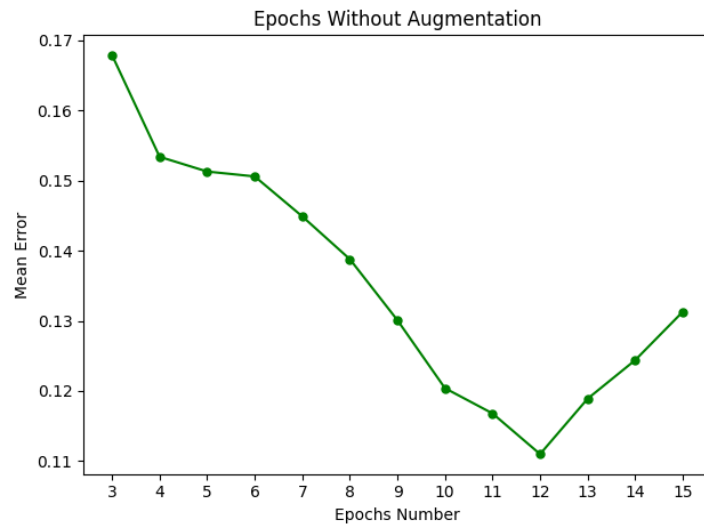




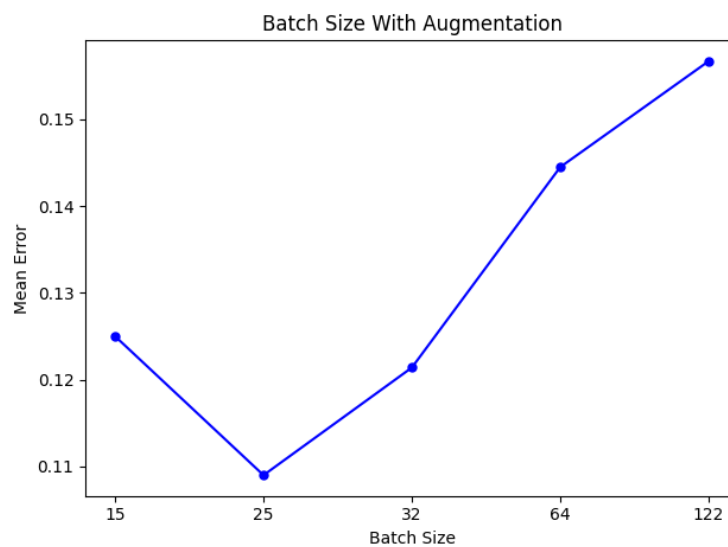
ניתן לראות כי התנהגות הגרף בשני המקרים היא זהה, הערך האופטימלי עבור הdecay הוא 0.1, ועל כן בחרנו בו.

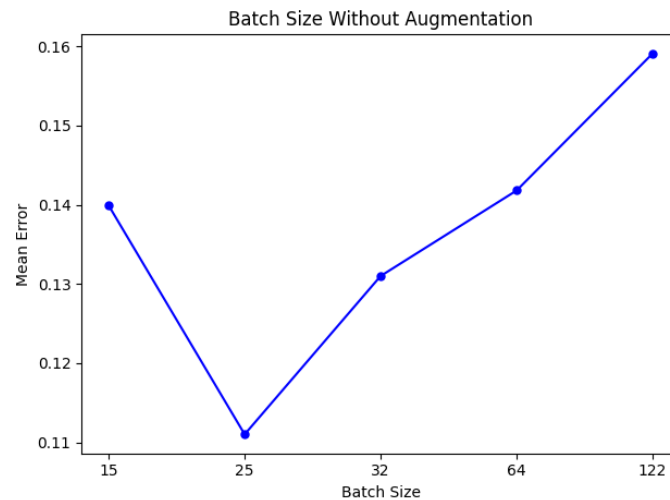
2. Epochs: פרמטר זה בעצם בא על חשבון תהליך האוגמנטציה באופן ישיר, מכיוון שככל שיש יותר תמונות, כך כמות ה Epochs שנוכל להריץ על הרשת מצטמצמת. תחילה כאשר הייתה לנו כמות גדולה של תמונות, הרצנו טווח ערכים שבין 1-5, ואף הגענו לתוצאות הגבוהות ביותר, למעלה מ-90 אחוז על סט הוולידציה והטסט אך חרגנו ממגבלת הזמן, והתוצאות עם 4 Epochs היו נמוכות מהתוצאות ללא אוגמנטציה, ועל כן, בחרנו לוותר עליה. טווח הערכים החדש הינו בין 3-15, הערך האופטימלי אליו הגענו הוא 12, וניתן לראות כי החל מנקודה זו אנחנו נכנסים למצב של over-fitting, המודל כבר העביר יותר מדי פעמים את התמונות ברשת ולמד אותם יתר על המידה.





3. Batch Size: גם פרמטר זה, בדומה לDecay, היה עם אותו טווח של ערכים בשני המקרים, ובשני המקרים הצביע על 25 כבחירה האופטימלית.





## המודל הנבחר וערכי הפרמטרים הנבחרים

לאחר האפטום הפרמטרים, הגענו לפרמטרים הבאים:

1. Width, Height – 224
2. Optimization Method – SGD
3. Decay – 0.1
4. Momentum – 0.9
5. Epochs - 12
6. Batch Size – 25
7. Early Stopping Criterion – 0.001

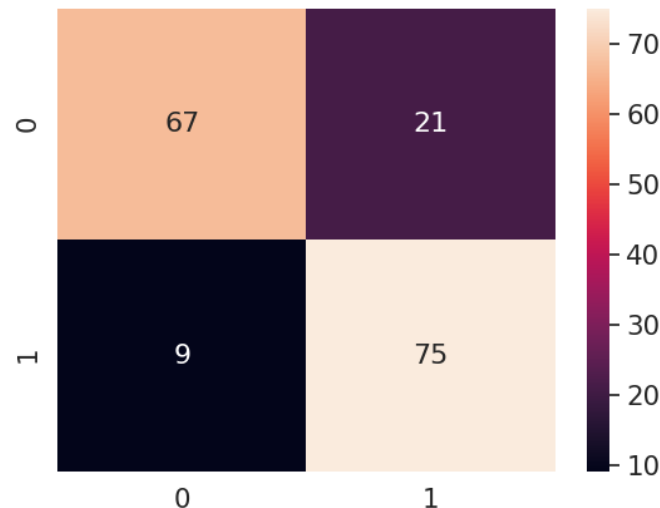


### שיפור המודל

תוצאות המודל הבסיסי על ה-82.5581% test:



:Confusion Matrix



שיפורי המודל:

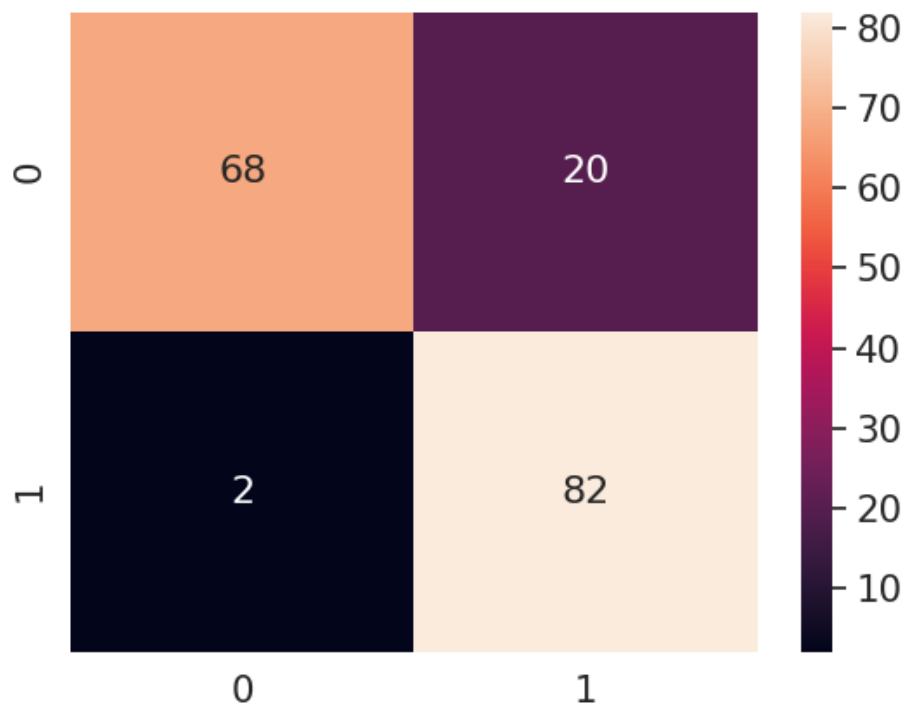
1. Weight Decay: כיוונו פרמטר זה, פורט בפרק הקודם.
2. Data Augmentation: בפרוצדורה זו, אנו מגדילים את הכמות של תמונות האימון שלנו, בעזרת הדאטה המקורי שניתן לנו. אנו מבצעים מניפולציות מסוימות על התמונות הקיימות (היפוך, הטיה, חיתוך וכו'), ובכך מגדילים את סט האימון שלנו, ויוצרים מודל מנוסה יותר, מבלי לאסוף דאטה חדש באמת. במשימה שלנו, סט האימון יחסית קטן, לכן אנו צופים ששיפור זה ישפר משמעותית את המודל שלנו. בנוסף, כמובן שאימון על סט גדול יותר של תמונות לוקח זמן רב יותר ואימון המודל מוגבל ל-25 דק'. כל מניפולציה שאנו מבצעים בעצם מגדילה את סט האימון בכמות התמונות הבסיסיות שקיבלנו, לכן נצטרך לבצע טרייד-אוף בין כמות המניפולציות שאנו רוצים לבצע אל מול הזמן הנדרש. בייחוד לאור העובדה שישנם עוד פרמטרים במודל זה אשר שיפור שלהם בא על חשבון זמן הריצה של המודל. לכן, בחרנו להכניס בקוד שלנו 5 מניפולציות שונות: Horizontal flip, Vertical flip, Rotate 90 Clockwise, Rotate 90 Counterclockwise, Crop. לאחר מכן, כאשר ראינו כי לא ניתן להכניס את כל הפרוצדורות במגבלת הזמן אשר ניתנה לנוף השתמשנו ב-3 הפרוצדורות אשר יוצרות את השיפור המשמעותי ביותר, ואת ה-2 הנותרות השארנו בהערה. בסופו של דבר, ראינו כי המודל האופטימלי מתקבל ללא שיפור זה, ועל כן לא השתמשנו בשיפור זה.
3. בנוסף, יש לציין כי נסינו שני מודלים בסיסיים שונים, (vgg16 ו mobilenet\_v2) אך בסופו של דבר, המודל הבסיסי הטוב ביותר היה ResNet50V2, ולכן השתמשנו בו.

### תוצאות המודל הסופי



תוצאות המודל הסופי על ה-87.2% test:








Confusion Matrix:



### שגיאות המודל

Second Type Error	First Type Error	Index
-------------------	------------------	-------



		1
0.9971775	0.99996257	
		2
0.7474469	0.9997018	
		3
	0.99925846	
		4
	0.9988369	
		5
	0.9987943	

[recall-precision curve](#)



גרף זה, מציג את הדיוק לעומת "קריאות השב", כפונקציה של threshold אשר מגדיר האם ההסתברות של סיווג התמונה 1 או 1-. כפי שאנחנו רואים, ככל שהthreshold יורד, כך מספר קריאות השב עולה. זאת מכיוון שככל שהוא נמוך, כך יותר תמונות יסווגו כפרח, ומצד שני, הסיכוי שלנו לטעות ירד.

