

## דו"ח עבודה 1 - למידה עמוקה

בדו"ח זה נציג את המחקר שלנו שנערך בשלושה שלבים על ה-Dataset שבחרנו Dog breed identification.

### שלב א' - ניתוח והבנת ה-Dataset

ה-Data מורכב משתי תיקיות המכילות תמונות מגוונות של גזעים של כלבים. תיקייה אחת שייכת ל-train set שבעזרתה אימנו את המודלים שיצרנו, ותיקייה נוספת ששייכת ל-test set עליה נפעיל את המודלים הטובים ביותר על מנת לקבל את התוצאות מאתר kaggle. המידע מגיע כתיקיית תמונות וקובץ csv שנותן לייבל לכל תמונה לפי שם התמונה.

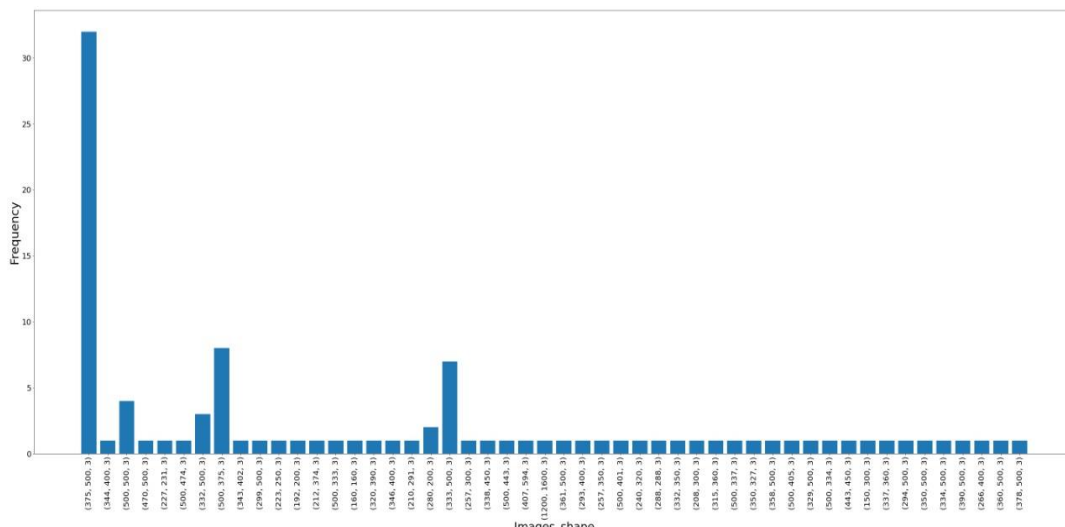
המטרה היא לזהות את סוג הכלב מתוך 120 כלבים.

### גודל ה-Data:

- מספר התמונות בתיקיית ה-train set הוא : 10222.
- מספר התמונות בתיקיית ה-test set הוא : 10357.
- מספר ה-labels (סיווגי סוגי הכלבים) הוא : 120.

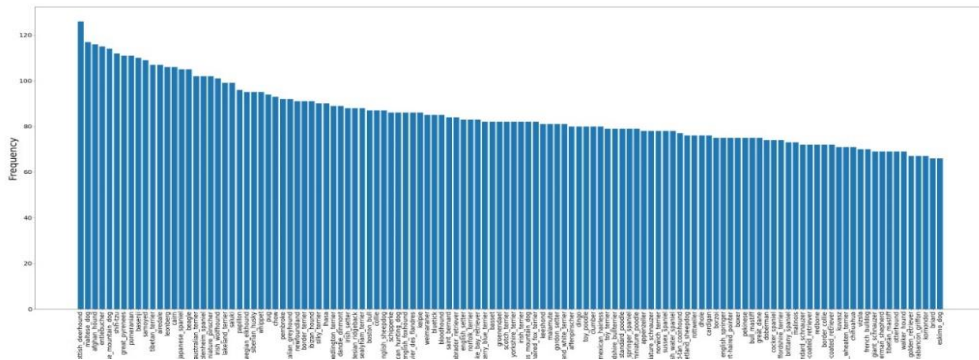
### מבנה התמונות:

- כל תמונה מורכבת משלושה ערוצים (RGB).
- התמונות לא מורכבות מאותם ממדים (מצורף גרף).
- לפי 100 התמונות בעלי הממדים הנפוצים ביותר, הממד הנפוץ ביותר הוא (375,500).



בעקבות נתונים אלו אנו מבינים כי אנו צריכים להמיר את התמונות לממד אחד משותף לכולם על מנת שנוכל להשתמש בהם במודל, בחרנו להשתמש בגודל של  $3 \times 128 \times 128$ .

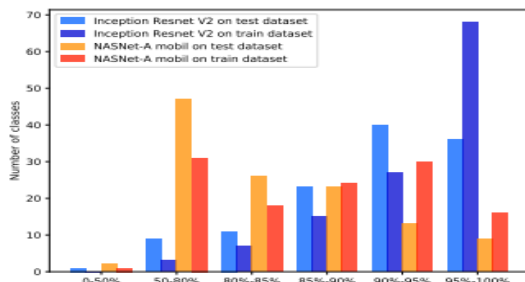
## אופן האיזון ב-Data:



מהגרף המצורף ניתן לראות שאומנם קיימים 120 labels אך הם מחולקים בצורה לא מאוזנת, כלומר אין מספר קרוב של תמונות עבור כל גזע של כלב ביחס לכלב אחר ולכן ישנה התפלגות שונה בכמות התמונות עבור כל סוג. סוג הכלב עם מספר התמונות הנפוץ ביותר מופיע למעלה מ-120 פעמים ואילו הכלב עם מספר התמונות הנמוך ביותר מופיע בערך 60-70 פעמים.

## קבלת אומדן לתוצאות – benchmark:

לאחר חיפושים באינטרנט של אתרים ומאמרים שעוסקים במחקרים ומודלים בנושא Dog breed identification ראינו שימוש במודלים שונים ומוכרים שהגיעו לתוצאות דיוק מרשימות של למעלה מ-90 אחוז.



לקוח מתוך מאמר: Dog Breed Identification Using Deep Learning.

קישור:

[https://www.researchgate.net/publication/328834665\\_Dog\\_Breed\\_Identification\\_Using\\_Deep\\_Learning](https://www.researchgate.net/publication/328834665_Dog_Breed_Identification_Using_Deep_Learning)

Table 1. Experimental Results of different Pre-trained Models

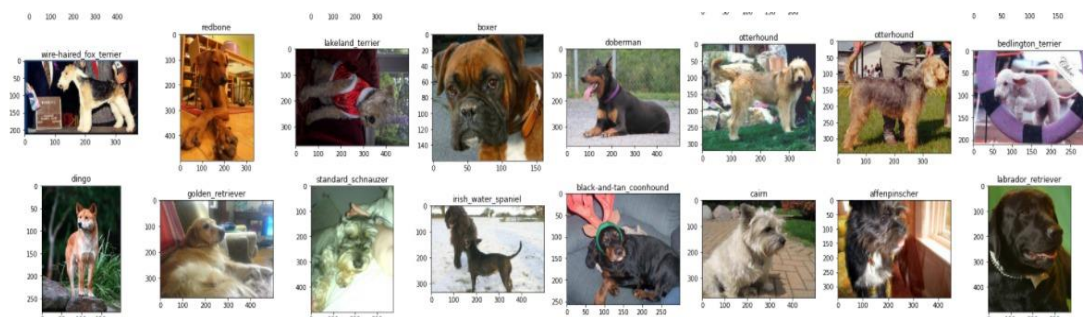
Model	Accuracy	Loss
VGG-16	91%	0.3
Inception V3	94%	0.06
Xception	93%	0.08

לקוח מתוך המאמר: Dog Breed Identification with Fine tuning of Pre-trained models

קישור:

<https://www.ijrte.org/wp-content/uploads/papers/v8i2S11/814640982S1119.pdf>

## דוגמאות לתמונות מה-Data:



מחלק התמונות שמוצגות ניתן לראות שיש תמונות שיחסית קל להבדיל בין סוגי הכלבים ויש כאלה שקשה יותר.

לדוגמא בין כלב מסוג boxer (שורה ראשונה תמונה רביעית משמאל) לבין כלב מסוג carim (שורה שנייה תמונה שישיית משמאל) קל יחסית לזהות ולהבחין בהבדלים, אך לעומת זאת בין

כלב מסוג carim (שורה שנייה תמונה שישית משמאל) לבין כלב מסוג affenpinscher (שורה שנייה תמונה שנייה מימין) קשה יותר לזהות את ההבדלים והם נראים די דומים.

### חלק ב'- בניית מודל התחלתי :

בשלב הראשון בנינו מודל התחלתי שמורכב כך :

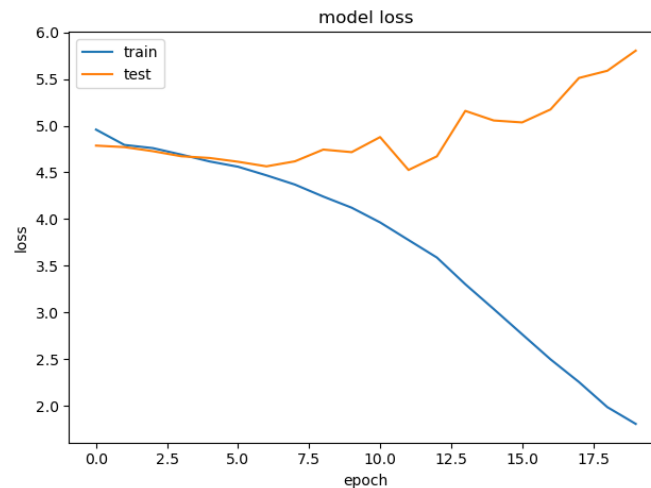
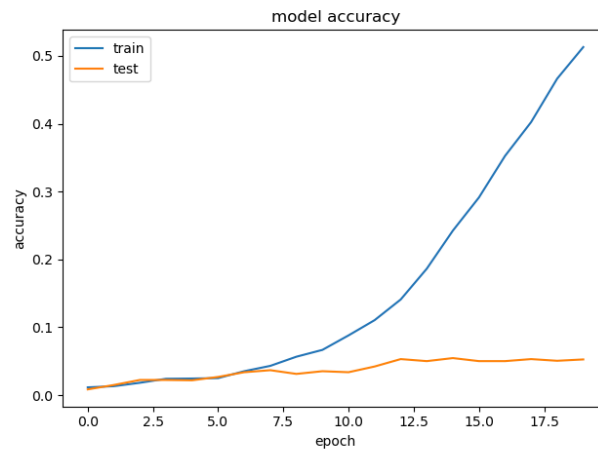
```
#32
x = Conv2D(64, 4, activation='relu', use_bias=True)(x)
x = Conv2D(64, 4, activation='relu', use_bias=True)(x)
x = MaxPool2D(pool_size=(2, 2))(x)
x = BatchNormalization()(x)
x = Dropout(0.15)(x)

# 32
x = Conv2D(128, 4, activation='relu', use_bias=True)(x)
x = Conv2D(128, 4, activation='relu', use_bias=True)(x)
x = MaxPool2D(pool_size=(2, 2))(x)
x = BatchNormalization()(x)
x = Dropout(0.15)(x)
```

מבנה בלוק במודל- בלוק 1,2  
64 מאפיינים, בלוק 3 128  
מאפיינים.

Model: "model"

Layer (type)	Output Shape	Param #
=====		
input_1 (InputLayer)	[(None, 128, 128, 3)]	0
conv2d (Conv2D)	(None, 125, 125, 64)	3136
conv2d_1 (Conv2D)	(None, 122, 122, 64)	65600
max_pooling2d (MaxPooling2D)	(None, 61, 61, 64)	0
batch_normalization (Batch Normalization)	(None, 61, 61, 64)	256
dropout (Dropout)	(None, 61, 61, 64)	0
conv2d_2 (Conv2D)	(None, 58, 58, 64)	65600
conv2d_3 (Conv2D)	(None, 55, 55, 64)	65600
max_pooling2d_1 (MaxPooling2D)	(None, 27, 27, 64)	0
batch_normalization_1 (Batch Normalization)	(None, 27, 27, 64)	256
dropout_1 (Dropout)	(None, 27, 27, 64)	0
conv2d_4 (Conv2D)	(None, 24, 24, 128)	131200
conv2d_5 (Conv2D)	(None, 21, 21, 128)	262272
max_pooling2d_2 (MaxPooling2D)	(None, 10, 10, 128)	0
batch_normalization_2 (Batch Normalization)	(None, 10, 10, 128)	512
dropout_2 (Dropout)	(None, 10, 10, 128)	0
flatten (Flatten)	(None, 12800)	0
dense (Dense)	(None, 512)	6554112
dropout_3 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 120)	30840
=====		
Total params: 7,310,712		
Trainable params: 7,310,200		
Non-trainable params: 512		

תוצאות המודל :

בשלב זה אימנו את המודל עד 25 אופקים ללא שימוש באוגמנטציה כלל רק כדי לקבל הערכה ראשונית של יכולות המודל שלנו, החלטנו להוסיף אלמנט של עצירה מוקדמת של המודל כדי לחסוך בזמן ריצה, אך עם מדד סבלנות יחסית גבוה של 5 אפוקים. בנוסף בחרנו להשתמש בגודל תמונה של  $128 \times 128$  כדי להוריד את כמות המשקולות שיש למצוא וכך שוב לקצר את זמן הריצה. בחרנו להשתמש ב-3 חזרות של בלוקים שמורכבים מ-2 שכבות קונבולוציה, מקס פולינג, נרמול ו-dropout.

ניתן לראות כי המודל שלנו נכנס למצב של התאמת יתר, המודל שלנו התחיל לאחר 7 אפוקים להשתפר משמעותית יותר טוב בנתוני האימון לעומת נתוני המבחן ולכן הגענו למסקנה שהוא נכנס להתאמת יתר לנתוני האימון. ניתן לראות שלמרות שהמודל כולל אלמנטים להורדת התאמת יתר כמו נרמול ו-dropout המודל נכנס להתאמת יתר משמעותית.

מניתוח הסיווגים ראינו כי המודל טועה גם בסיווגים בהן ההסתברות המתקבלת היא גבוהה וגם בסיווגים בהן ההסתברות המתקבלת היא נמוכה.

בשלב הבא החלטנו להוסיף למודל שיטות אוגמנטציה שונות, מניתוח התמונות שביצענו ניתן לראות כי לרוב הכלב נמצא במרכז התמונה ולכן בחרנו לבצע הזזות והיפוכים לתמונה. נשתמש באוגמנטציה על התמונות על מנת שהמודל שלנו יוכל לזהות את הסיווג של התמונה גם בצורות שונות בהם תהיה התמונה וכך ננסה למנוע overfitting. מדובר ב-dataset המורכב מתמונות של כלבים לכן בעזרת אוגמנטציה נוכל להשתמש במספר אפשרויות על התמונות כגון: סיבוב של

התמונה לפי מעלות, הזזה של התמונה לאורך או לרוחב, זום אין/ אוט ועוד. יש אפשרויות באוגמנטציה שלא נשתמש בהם על תמונות אלו כמו למשל סיבוב אנכי כי כך התמונה תאבד את המשמעות שלה לעומת תמונות אחרות ואז הסיכוי לסיווג לא נכון יגדל.

שיטות האוגמנטציה אותן בחרנו להוסיף בשלב זה :

- Horizontal flip
- Rotation range
- Width shift
- Channel shift
- Height shift
- zoom

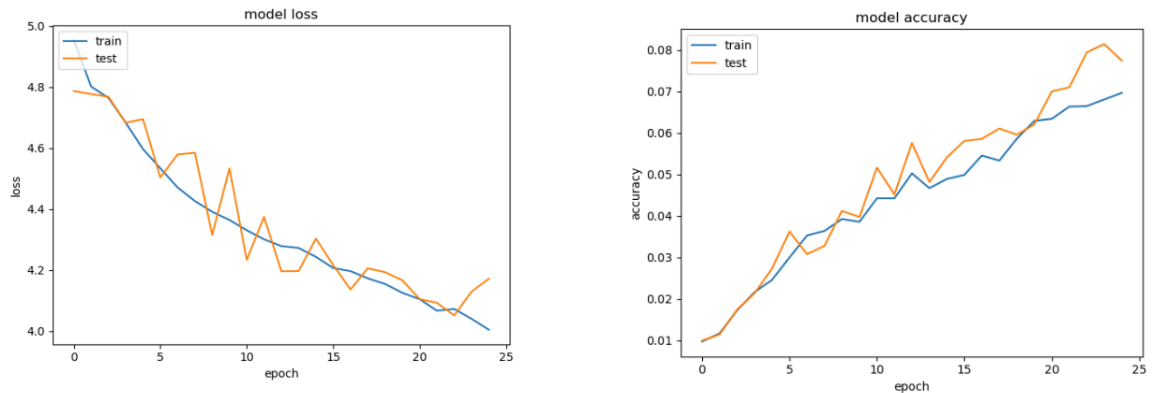
בנוסף לאלמנט של אוגמנטציה בחרנו להשתמש גם באלמנט של רגולריזציה עבור השכבות הצפופות של הרשת, בחרנו להשתמש ברגולריזציה מסוג l2 מכיוון שהיא מורידה את משקלי המשקולות ולא מאפסת אותם כמו ב-l1 ואחרי שקראנו שלרוב היא יותר פופולארית.

בחרנו להשתמש במקדם של 0.01.

אלמנט נוסף שחשבנו להשתמש בו אבל בשלב זה החלטנו לוותר עליו הוא העלאת ה-dropout. בחרנו לא להוריד את מורכבות המודל על ידי שינוי הארכיטקטורה מכיוון שהמודל שלנו לא הצליח לעלות את רמת הדיוק על נתוני האימון ביותר מכמה אחוזים בודדים.

## מודל מספר 2- הוספת אוגמנטציה ורגולריזציה

### תוצאות המודל :



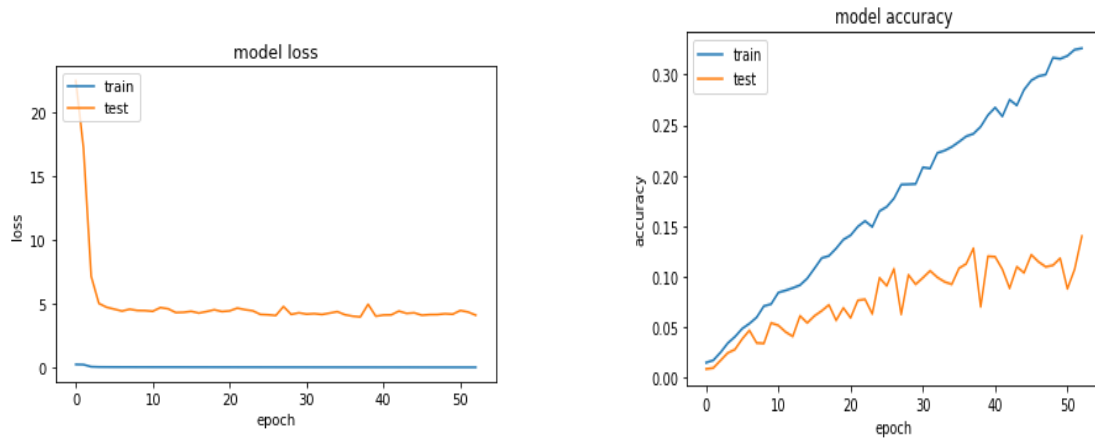
ניתן לראות כי מודל זה לא נכנס להתאמת יתר במהלך כל שלב האימון והצליח להגיע לדיוק של כמעט 8% בנתוני המבחן. אנחנו רואים שאוגמנטציה ורגולריזציה עזרו לנו לצמצם משמעותית את ההתאמת יתר. חשוב להוסיף כי זה המודל הטוב ביותר שיצא מבין חמש ההרצות שביצענו בעזרת Stratified k-fold (על מנת לשמור על יחס זהה בין המחלקות בנתוני האימון ונתוני המבחן) ויתר התוצאות היו נמוכות יותר והיה נראה כי המודל לרוב מתקשה לעבור את האחוזים הבודדים (3%-5%).

מניתוח תוצאות המודלים ראינו כי הכלבים שהמודל צודק בהם הם הכלבים שנבחרים הכי הרבה פעמים על ידו, בנוסף ראינו כי הגיוון בתוצאות של הסיווגים הוא גם כן איננו גבוה ולרוב הוא מסתכם בכך שהמודל תמיד בוחר רק שליש מסוגי כלבים בלבד מתוך 120 ויש חלוקה מאוד לא מאוזנת בבחירות שלו. מכאן הסקנו שאנחנו צריכים לייצר עוד תמונות של כלבים שלהם יש מספר נמוך של תמונות, לאמן את המודל לאורך יותר זמן על מנת לראות אם הוא ממשיך להשתפר ואם זה לא יעזור אז נצטרך לעלות את מורכבות המודל.

הכלבים שעבורם המודל מדויק (עמודה G ערך 1), הם הכלבים שהמודל בוחר הכי הרבה פעמים (עמודה H). עמודה I,H מסמנות לייבל וכמה פעמים הוא נבחר וניתן לראות שזה תואם לכלבים שהמודל חזה נכון.

I	H	G	F	E	D	C	B
predictions	repeat	accuracy	real_pred	predictions	prediction	true_label	id
whippet	6	1	0	whippet	0.014282	whippet	9513287b3
scottish_deerhound	228	1	0	scottish_deerhound	0.014877	scottish_deerhound	01430d1dd64
newfoundland	97	1	0	scottish_deerhound	0.015883	scottish_deerhound	0057c9ab97
maltese_dog	110	1	0	scottish_deerhound	0.017149	scottish_deerhound	0669d0c296
irish_water_spaniel	260	1	0	newfoundland	0.021769	newfoundland	2efc4b984e
cairn	248	1	0	newfoundland	0.021894	newfoundland	befc8bac9
bouvier_des_flandres	184	1	0	maltese_dog	0.017697	maltese_dog	fa5054c51
black-and-tan_coonhound	170	1	0	irish_water_spaniel	0.093717	irish_water_spaniel	52e4bb5c9
bernese_mountain_dog	188	1	0	irish_water_spaniel	0.043529	irish_water_spaniel	37f52e1a
australian_terrier	355	1	0	irish_water_spaniel	0.053091	irish_water_spaniel	c9c0e4d1f
blenheim_spaniel	10	1	0	cairn	0.016843	cairn	b83be4ce
leonberg	34	1	0	bouvier_des_flandres	0.024885	bouvier_des_flandres	deb8536d0f
schipperke	15	1	0	black-and-tan_coonhound	0.035883	black-and-tan_coonhound	3e2f22697
groenendael	41	1	0	black-and-tan_coonhound	0.043235	black-and-tan_coonhound	833c2b66f
beagle	3	1	0	black-and-tan_coonhound	0.034888	black-and-tan_coonhound	71c10b10f
basenji	8	1	0	bernese_mountain_dog	0.13549	bernese_mountain_dog	n1241cd8dc
samoyed	4	1	0	bernese_mountain_dog	0.028892	bernese_mountain_dog	n52fc29c92
pomeranian	3	1	0	bernese_mountain_dog	0.085224	bernese_mountain_dog	na064e199c
airedale	11	1	0	australian_terrier	0.014174	australian_terrier	2ea0a5df9
japanese_spaniel	9	1	0	australian_terrier	0.014174	australian_terrier	6aae9f70d
irish_wolfhound	7	1	0	australian_terrier	0.014174	australian_terrier	94134ebef
shih-tzu	5	1	0	australian_terrier	0.014174	australian_terrier	aad0dede6t
sealyham_terrier	8	0	77	australian_terrier	0.014174	australian_terrier	001c401b
tibetan_mastiff	10	0	61	bernese_mountain_dog	0.034291	bluetick	00214f311
curly-coated_retriever	5	0	69	cairn	0.024975	walker_ho	0067dc3ee
labrador_retriever	2	0	73	australian_terrier	0.014174	bluetick	006cc3ddt
irish_setter	1	0	59	australian_terrier	0.014174	otterhound	009509be3
boston_bull	2	0	96	black-and-tan_coonhound	0.023224	dingo	00a338a9c
tibetan_terrier	2	0	28	irish_water_spaniel	0.073028	standard	00a86239c
bedlington_terrier	3	0	36	australian_terrier	0.013753	weimaraner	00fa64131
afghan_hound	7	0	10	australian_terrier	0.014318	groenendael	01e87fd2f
norwegian_elkhound	4	0	107	irish_water_spaniel	0.067321	hole	011e0676c
scotch_terrier	1	0	43	australian_terrier	0.014329	german_shepherd	010180c0f
toy_terrier	1	0	41	blenheim_spaniel	0.026485	greater_springer	0177a92af
wire-haired_fox_terrier	1	0	31	bernese_mountain_dog	0.038467	miniature	01b6c2bd3
miniature_pinscher	1	0	105	bouvier_des_flandres	0.030504	brittany	01b849a7e
border_terrier	1	0	107	australian_terrier	0.014174	english_setter	01c230ec1

המודל הטוב ביותר שהגענו אליו היה כאשר השתמשנו בגודל תמונה של  $64 \times 64$  והגענו לדיוק של 15% אבל עם התאמת יתר גבוה ולאורך מספר אפוקים גבוה יותר.



השתמשנו ב- stratified cross validation כדי שהחלוקה תהיה שווה בין אימון למבחן.

טבלת השוואה בין מודלים :

model	Time per epoch	Accuracy	Loss	Hyper parameters	metrics	conclusions	Features add	Cross validation
1	44 דק	3%	4.8	שכבות קונבולוצי ה-64 פילטרים בשתי השכבות הראשונות 128 ו1 בשכבת קונולוציה אחרונה. Dropout של 15%, מקס פולינג $2 \times 2$	Dropout, early stoping, batch normalization	המודל נכנס להתאמת יתר,		
2	44 דק	8%	4.1	0.01	אוגמנטציה, רגולריזציה L2 וסעיף קודם.	המודל צודק בכלבים שהוא בוחר הכי הרבה.	אוגמנטציה, רגולריזציה	דיוק – 4.2% במוצע, הפסד – 4.8% במוצע

המלצות לשיפור המודל :

- ריצה עם מספר אפוקים גדול יותר כי המודל המשיך לטפס בדיוק האימון והמבחן.
- הוספת מורכבות למודל .
- הוספת אוגמנטציה לסוגי כלבים עם מעט תמונות על מנת להגדיל את נתוני המבחן ולגרום לאיזון בבחירת סוגי הכלבים.

## שלב ג' – ביצוע Transfer Learning :

בשלב זה התבקשנו לבחור מודל קיים מאומן כדי להשתמש בו ולהפעיל אותו למשימת החיזוי שלנו.

בחרנו לקחת את ארכיטקטורת inception v3 לאחר שקראנו במאמרים כי היא ארכיטקטורה יעילה לסיווג תמונות ומביאה תוצאות דיוק גבוהות מאוד.

Layer (type)	Output Shape	Param #
inception_v3 (Functional)	(None, 5, 5, 2048)	21802784
flatten (Flatten)	(None, 51200)	0
dense (Dense)	(None, 512)	26214912
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131328
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 120)	30840
Total params: 48,179,864		
Trainable params: 26,377,080		
Non-trainable params: 21,802,784		

הוספנו 6 שכבות למודל ה-inception v3 המקורי כדי שלבסוף המודל שלנו יגיע לסיווג של 120 לייבלים.

בחרנו לקחת את השכבה האחרונה מהמודל כיוון שזה סוף הבלוק במודל, למדנו שמודל זה בנוי מבלוקים שחוזרים על עצמם ורצינו לקחת את השכבה האחרונה שמהווה את סוף הבלוק שלה עם המשקלים המעודכנים ביותר. השכבה האחרונה היא (5,5,2048), ביצענו לאחר מכן Flatten כדי להביא את המידע למימד אחד ולאחר מכן הוספנו שכבות Dense ו-DropOut.

ביצענו data augmentation (כדי למנוע overfitting ולהעשיר את סט האימון) עם הפרמטרים הבאים:

```
rotation_range=18,
rescale=1./255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True,
fill_mode='nearest',
width_shift_range=0.1,
height_shift_range=0.1,
```

- רוטציה של תמונה – סיבוב לשני הכיוונים ב-18 מעלות
- הקטנת/הגדלת התמונה כדי שהתמונות יהיו באותו גודל
- דיסטורציה לתמונה כדי לשנות את הזווית של התמונה כולה
- הפעלת זום למרכז התמונה
- הפיכת הכיוון של התמונה בכיוון האופקי
- השלמת מסגרת תמונה בצבעים הכי קרובים
- הזזת תמונה לרוחב
- הזזת תמונה לאורך

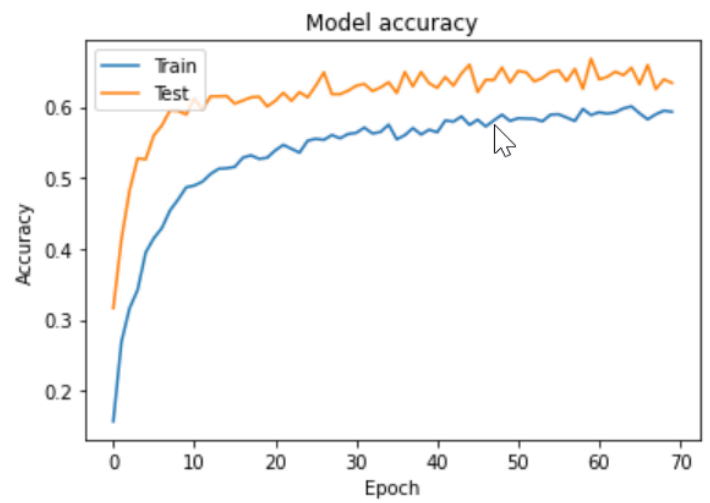
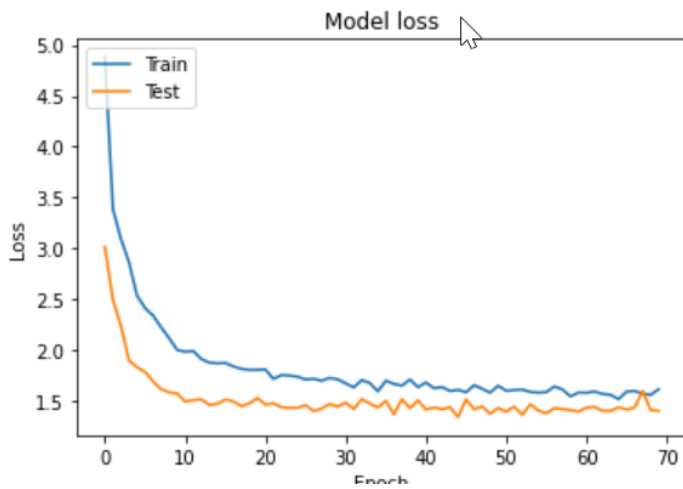
ואימנו את סט האימון על 80% מהרשומות והשארנו 20% לסט הוולידציה.

לאחר ריצה של 70 epochs קיבלנו תוצאת דיוק על סט האימון של 60.09% עם loss של 1.5206.

על סט הוולידציה קיבלנו תוצאת דיוק של 62.87% עם loss של 1.549.

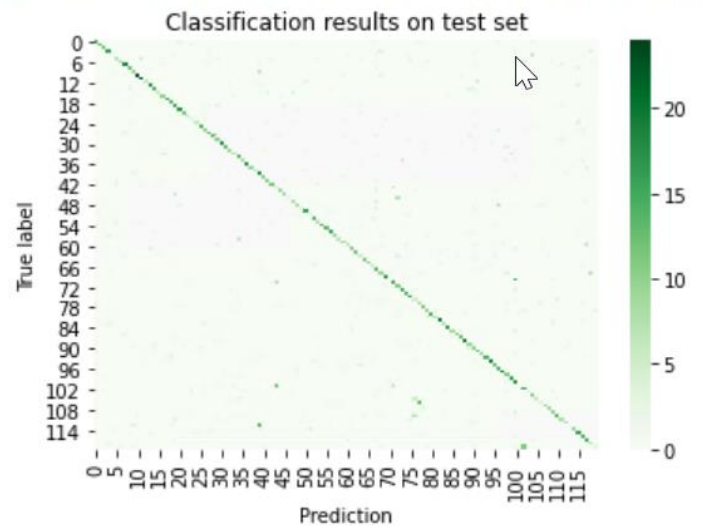


## תוצאות המודל:

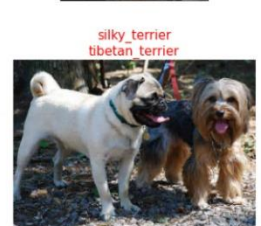


model accuracy on test set is: 62.866927592954994%

```
[[13  0  0 ...  0  0  0]
 [ 0 14  0 ...  0  0  0]
 [ 0  0 11 ...  0  0  0]
 ...
 [ 0  0  0 ... 14  0  0]
 [ 0  0  0 ...  0  7  0]
 [ 0  0  0 ...  0  0  2]]
```



## מדגם של סיווג כלבים של המודל:



ניתן לשים לב שהמודל לא הצליח לסווג כמו שצריך את הכלב מסוג Silky\_terrier והתבלבל עם Tibetan\_terrier – שני הכלבים הם סוג של גזע הטרייר ולכן אפשר להבין את אי ההצלחה בקטגוריה זו.

```
[[12  0  0 ...  0  0  0]
 [ 0 16  0 ...  0  0  0]
 [ 0  0 11 ...  0  0  0]
 ...
 [ 0  0  0 ... 15  0  0]
 [ 0  0  0 ...  0  6  0]
 [ 0  0  0 ...  0  0 11]]
```

בהמשך, ביצענו feature extractor בעזרת המודל הנ"ל (מודל ה-transfer learning) - הורדנו מהמודל את השכבה האחרונה וביצענו בעזרתו פרדיקציה לסט האימון.

לאחר ביצוע חיזוי על סט הוולידציה קיבלנו דיוק של 65.8% - שיפור של 3% ממודל ה-inception v3:

model accuracy on validation set is: 65.8023483365949%



transfer learning הביא לשיפור משמעותי בסט הבדיקה, שהרי ידוע כי רשתות קונבולוציה מחזיקות בייצוג רחב של הפיצ'רים בקלט מה שעוזר להגיע לתוצאות דיוק טובות יותר, מודלי למידת המכונה הקלאסיים כדוגמת רגרסיה המשתמשים במשקולות שנלמדו ע"י רשת קונבולוציה יכולות לשפר את התוצאות כמו במקרה זה.

מודל	זמן ריצה	loss	היפר-פרמטרים התחלתיים	מטריקות בהן השתמשנו	דיוק על סט הוולידציה
Inception v3	205 דק'	1.549	<p>ביצירת המודל –</p> <pre>include_top = False, weights = 'imagenet', input_shape = (224,224)</pre> <p>שכבות שנוספו –</p> <pre>Flatten(), Dense(512,activation = 'relu'), Dropout(0.3), Dense(256,activation = 'relu'), Dropout(0.3), Dense(120,activation = 'softmax')</pre> <p>קמפול המודל –</p> <pre>loss = 'categorical_crossentropy', optimizer = Adam(), metrics = ['accuracy']</pre>	<p>מחבילת Sklearn.metrics</p> <p>השתמשנו ב:</p> <pre>accuracy_score, log_loss, confusion_matrix</pre>	62.87%
Logistic regression	32 דק'		<pre>multi_class='multinomial', solver='lbfgs', n_jobs=8</pre>		65.8%

#### סעיף ד- סיכום ומסקנות:

- ראינו שאוגמנטציה עוזר להוריד את ההתאמת יתר בורה משמעותית.
- מודל שנבנה על ידנו הוא הרבה יותר קשה לשיפור מאשר שימוש ב-transfer learning.
- הבנת איפה המודל טועה ואיזה סוגי טעויות הוא מבצע, עוזר לשפר את המודל בצורה טובה מאשר שינוי של הארכיטקטורה של המודל.
- יכולת המחשב שלנו לא מסוגלות להריץ מודל מורכב מדי ולכן נאלצנו להסתפק במודלים יחסית קלים.
- היה כיף להתנסות בפעם הראשונה בבעיה אמיתית זיהוי תמונה ולהתמודד עם כל הקשיים כמו הבנת הבעיה, שיפור מודלים והתנסות בלמידה עמוקה.