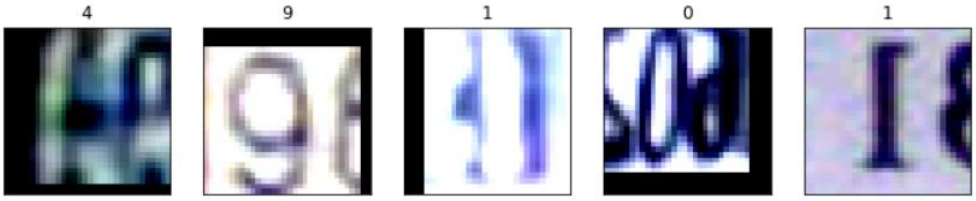
דו"ח סיכום לתרגיל 2 בראייה ממוחשבת
נושאים: Deep Neural Networks, Classifiers & Features
תאריך:11.05.20

מגישים:
אופק חררי 305199879
שני ישראלוב 204679153

הערה: עבדנו ב- google Collaboratory notebooks, השתמשנו בקבצים שניתנו בתרגיל וגם יצרנו תיקיית our_images שמכילה 5 תמונות של בעלי חיים כפי שנדרשנו בתרגיל (1 לסעיף 5 ו-4 לסעיף 10), התיקייה מצורפת עם מחברת ה-google collab.

Part 1 – design and build a CNN Classifier

1.1 טעינת ה-SVHN data set והצגת 5 תמונות מתוך ה-train set:



1.2 תיאור תכנון ה-CNN לסיווג ספרות מהתמונות:

תיאור הארכיטקטורה:

- 3 שכבות קונבולוציה ושכבת FC.
- הגדלים של ה-filter: בכל שכבות הקונבולוציה השתמשנו בפילטרים בגדלים 3x3. מספר הערוצים נקבע לפי מספר הערוצים שרצינו ביציאה.
- השתמשנו ב-fully-connected layer , שכבת ה-FC בוצעה לאחר שעשינו "שיטוח" למוצא של שלושת שכבות הקונבולוציה. בכניסה השכבה קיבלה וקטור באורך 4096 וביציאה 10.
- השתמשנו ב-max pooling כדי להקטין את המימדים, ובפונקציית dropout כדי ליצור ארכיטקטורה יותר חסונה.
- מימדי הקלט הוא תמונה ברוחב ואורך 32x32 ובעלת 3 ערוצי צבע
- מימדי הפלט הוא וקטור של 10 מחלקות, כאשר כל מחלקה מתארת ספרה ואת ההסתברות שזו המחלקה שמתארת את התמונה הנתונה.
- מספר הפרמטרים (משקולות) ברשת:

$$weights = 864 + 18432 + 73728 + 147456 + 294912 + 58824 + 4194304 + 524288 + 5120 = 5{,}848{,}928$$

כחול – המשקלים משכבת קונבולוציה ראשונה, כתום – שנייה, ירוק – שלישית, צהוב – FC.
הערה: יצרנו תרשים של הארכיטקטורה שבנינו שמופיע בסוף הדו"ח.

1.3 אימון המסווג

- **תאר את ה-hyper-parameters**
  - Batch size הוא 64
  - Epochs הוא 20
  - Learning rate הוא 1e-4
  - Validation set - לא השתמשנו, בדקנו היפר-פרמטרים בעצמנו.

- **מה הדיוק הסופי על ה-test set**

```
Epoch: 17 | Loss: 0.1693 | Training accuracy: 95.913% | Test accuracy: 94.741% | Epoch Time: 85.06 secs
Epoch: 18 | Loss: 0.1648 | Training accuracy: 96.160% | Test accuracy: 94.430% | Epoch Time: 85.11 secs
Epoch: 19 | Loss: 0.1579 | Training accuracy: 96.219% | Test accuracy: 94.541% | Epoch Time: 85.77 secs
Epoch: 20 | Loss: 0.1535 | Training accuracy: 96.362% | Test accuracy: 94.603% | Epoch Time: 85.15 secs
```

הערך על ה-epoch האחרון מספר 20 הוא 94.603%. אחוז דיוק מספק.

Part 2 - Analyzing a Pre-trained CNN

2.1 טעינת VGG16
2.2 התמונות בתיקיית birds

bird_0.jpg

bird_1.jpg

2.3 עיבוד מקדים על התמונות כדי שיתאימו לארכיטקטורת VGG16

יצרנו פונקציה preprocess מקבלת את התמונה ומבצעת עליה מספר פעולות:
Resize(224,224) – הפונקציה הזאת מתאימה את התמונה לגדלים של ארכיטקטורת VGG16.
CenterCrop(224) – הפונקציה הזאת מחזירה תמונת PIL במימד הנתון.
ToTensor() – הפונקציה הזאת ממירה תמונות PIL או מערכים ל-tensor בתצורה (C x H x W) והטווח ערכים נע בין 0 ל-1.
Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) – הפונקציה הזאת מנרמלת את התמונת tensor לפי תוחלת ווריאנס שנתונים בתמונות של הארכיטקטורה.
לאחר מכן אנחנו מבצעים unsqeeze כדי ליצור input_bach. זזה יוצר mini-batch כפי שמצופה מהמודל.

2.4 feed למודל
מה הם ה-outputs?

```
bird_0.jpg
calssification : hummingbird, with probability 70.39%
bird_1.jpg
calssification : lorikeet, with probability 64.63%
```

ההתאמה ל-bird_0 היא נכונה (ע"פ הסתכלות אנושית שלנו) ואחוזי הסתברות יחסית גבוהים.
ההתאמה ל-bird_1 היא לא מדוייקת אך קרובה.

2.5 מצא תמונה של ציפור\חתול\כלב באינטרנט, הצג והכנס למודל

מצאנו תמונה של כלב באינטרנט והכנסנו למודל. התמונה מופיעה בתיקיית our_images שצירפנו לתרגיל.

מה ה-outputs?

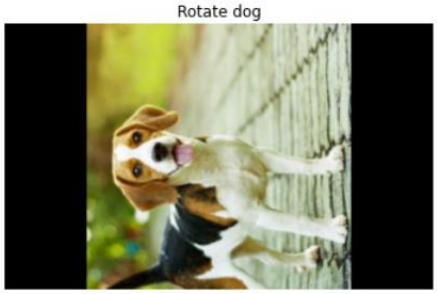
```
calssification : English_foxhound, with probability 46.12%
```

our_dog.JPG

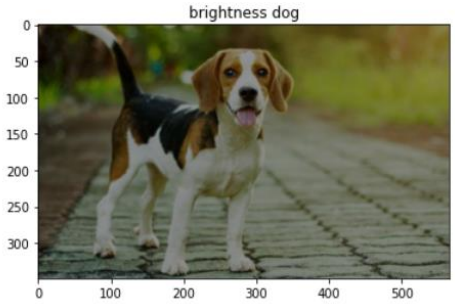
2.6 בצעו שלוש התמרות על התמונה מסעיף 5 והציגו אותן

התמונות אחרי הטרנספורמציות:

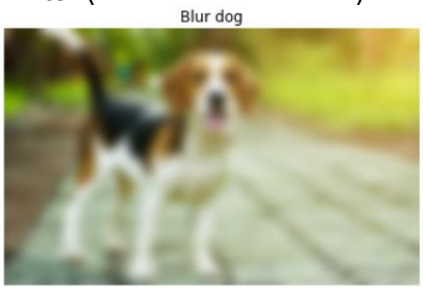
- סיבוב: geometric transformation



- בהירות: color transformation



- פילטר: (בחרנו טשטוש גאוסיאני) filter



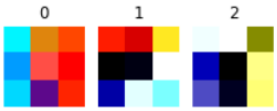
2.7 אחרי feed לרשת, מה ה-output ? מה שונה מסעיף 5?

```
rotated
calssification : Chihuahua, with probability 18.76%
brightness
calssification : Walker_hound, with probability 44.40%
GaussianBlur
calssification : rock_beauty, with probability 9.17%
```

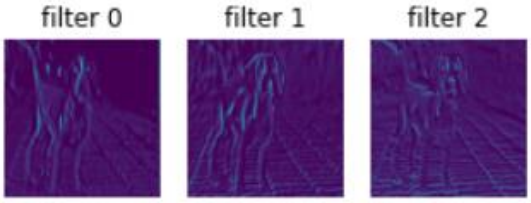
בסעיף 5 קיבלנו סיווג בהסתברות של 46.12% ומבדיקה אנושית שלנו הסיווג למחלקת English_foxhound הוא נכון, בתמונה המסובבת קיבלנו ירידה של כ-27% בהסתברות וגם הכלב שבחרנו לא דומה לכלב שניתן בסיווג. בתמונה שבה שינינו בהירות קיבלנו אחוזים מאוד דומים לתמונה המקורית וגם הסיווג מאוד דומה. בתמונה המטושטשת קיבלנו את האחוזי הסתברות הכי נמוכים וקיבלנו דג ולא כלב. נוכל להסיק שהרשת כנראה חסינה יחסית בפני סיבובים, וחסינה בפני שינויי צבע, אך לא לרעשים חזקים.

2.8 שלושת הפילטרים בשכבה הראשונה ב-VGG16

אלה הן המשקולות של שלושת הפילטרים בשכבה הראשונה ב- VGG16

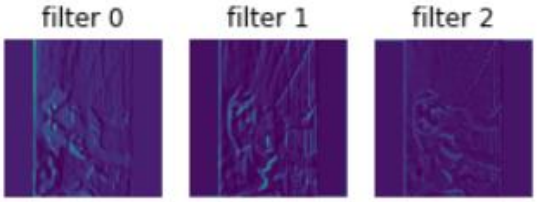


הפלט של הפילטרים לתמונה מסעיף 5 (הכלב המקורי)

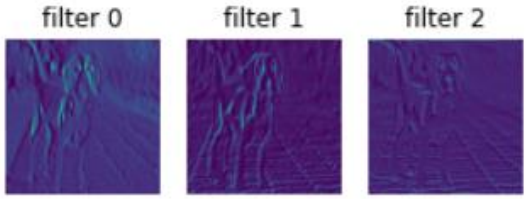


הפלט של הפילטרים לשלושת התמונה מסעיף 6:

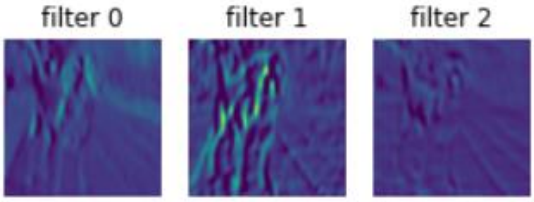
- הכלב המסובב



- הכלב עם שינוי בהירות



- הכלב עם טשטוש גאוסי



רואים את השימוש של שלושת הפילטרים על התמונות בשכבה הראשונה, קיבלנו קווי מתאר של התמונות.

2.9 לכל תמונה בתיקיית ה- dogs, cats שמור את ה-feature vector שלהם משכבת FC.

איזו שכבה בחרתם?

שכבה מספר 7.

ניתן לראות את ההתייחסות לבחירת השכבה בקטע קוד הרלוונטי לתרגיל.

```
for layer in modulelist[1:7]:
```

מה הגודל של ה-feature space?

הגודל של ה-feature vector הוא 4096 על 20 תמונות (10 כלבים, 10 חתולים).

2.10 מסווג SVM כדי לסווג חתולים וכלבים בהתבסס על ה-feature vector.

השתמשנו ב-4 תמונות מהאינטרנט כ-test set , התמונות הללו מופיעות בתיקיית our_images שמצורפת לתרגיל.

מה שם האלגוריתם?

השתמשנו באלגוריתם SVM.

מהן התוצאות?

להלן התוצאות. ניתן לראות שהמסווג זיהה נכונה את הכלבים והחתולים.

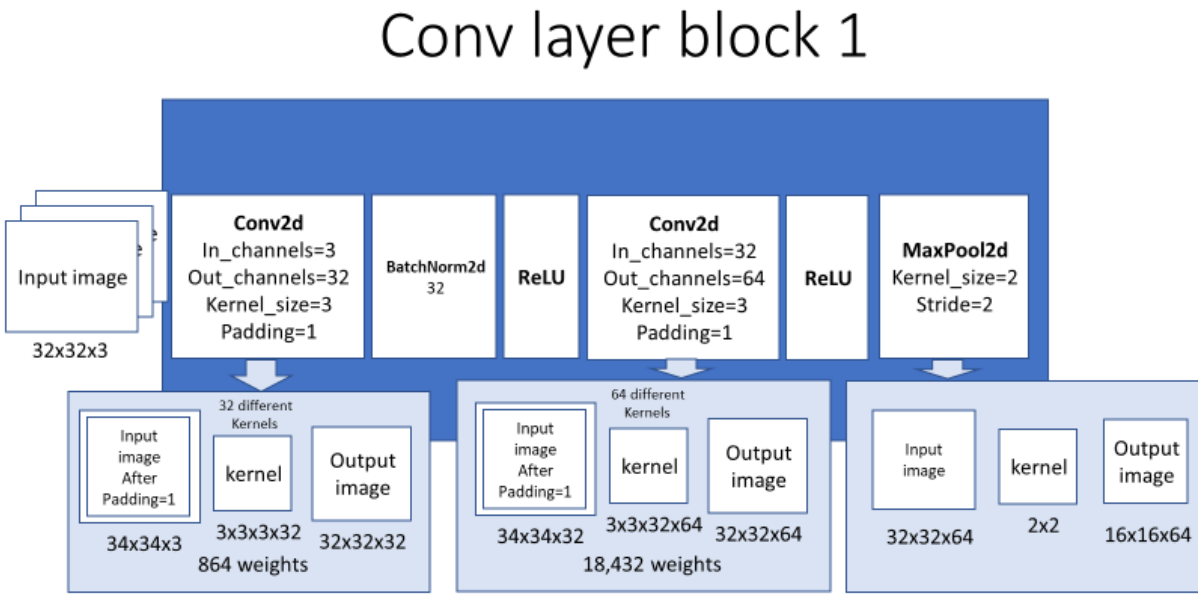
Dog

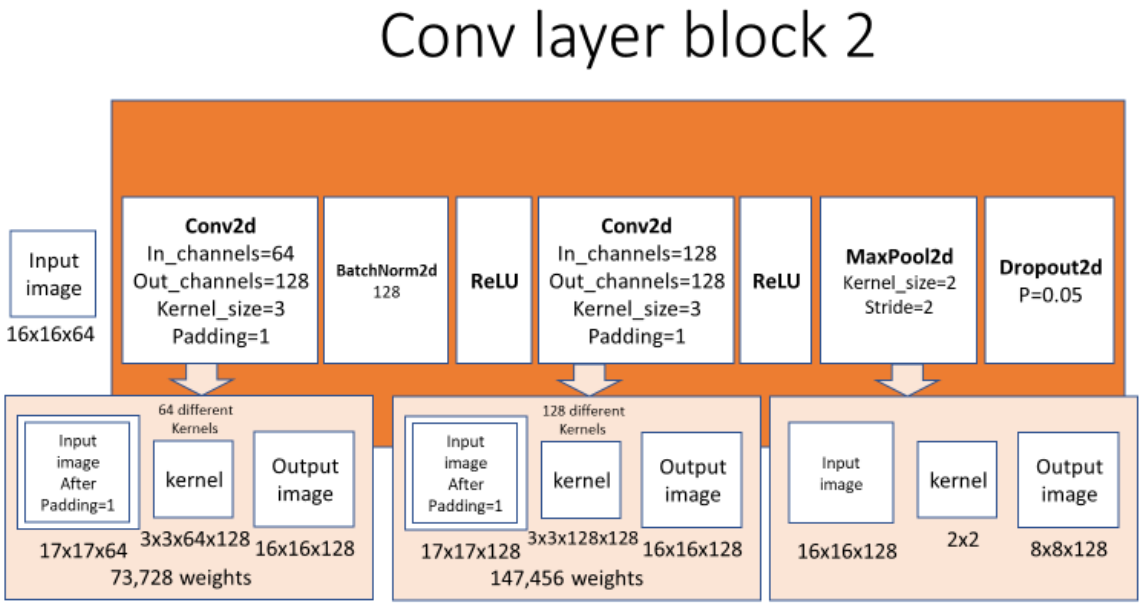
Dog

Cat

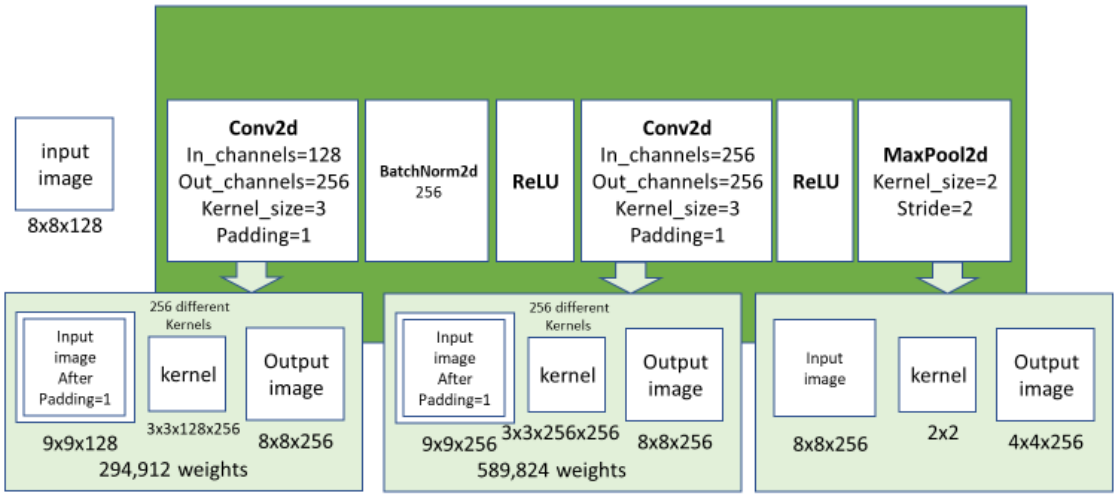
Cat

נספח ל- 1.2 תיאור תכנון ה-CNN לסיווג ספרות מהתמונות:





Conv layer block 3



Fully connected layer