

Carnet

מגישים: נתנאל לב & איתן קוסמן

מנחים: רון סלוסברג & ירון חונן

דצמבר 2018



תוכן העניינים

1	תוכן העניינים
2	מבוא ותקציר
3	תיאור המערכת
	תיאור הביצוע
5	איסוף נתונים
6	עיבוד נתונים
10	תהליך האימון
12	תוצאות
14	מסקנות והמלצות



מבוא ותקציר

בפרויקט זה פיתחנו מערכת end-to-end מבוססת רשת CNN לנהיגה אוטונומית בסימולטור. הפרויקט מבוסס על מאמר מאת חברת Nvidia. הקלט למערכת מגיע ממצלמה הממוקמת על שמש הרכב, והפלט הוא פקודת היגוי.

קיימת גישה לבניית מערכת אוטונומית הדוגלת בחלוקה לתתי משימות. בתחום הרכב האוטונומי משימות אלה בדרך כלל יכללו זיהוי נתיב נסיעה, תכנון מסלול ובקרה. קיימת אסכולה המתנגדת לגישה זו וטוענת שחלוקה זו היא מלאכותית. הטיעון המרכזי הוא שחלוקה זו נבחרה כי היא מקלה על בני אדם לגשת לבעיה, אך כלל לא מובטח שהיא תביא לביצועים אופטימליים תוך שימוש בכלי למידה עמוקה.

גישת end-to-end דוגלת בכך שניתן להגיע לתוצאות טובות יותר ללא חלוקה לתתי משימות, על ידי יצירת רשת במבנה המתאים לבעיה והזנתה במידע איכותי. כך תבצע אופטימיזציה פנימית המכוונת לביצועי המערכת כמכלול, בניגוד לחלוקה לתתי משימות ואופטימיזציה של כל משימה בנפרד. המערכת לומדת לפתור את הבעיה במספר צעדים מינימלי וכך נוצרות רשתות קטנות ומהירות יותר.

במהלך הפרויקט אספנו מידע (הקלטות נהיגה) ועיבדנו אותו כדי שיתאים לאימון. לאחר מכן מימשנו את הרשת המתוארת במאמר ואימנו אותה. האימון התבצע על סמך הקלטות נהיגה הכוללות צילומים ממצלמות הממוקמות בקדמת הרכב ואת זווית ההיגוי המתאימה לכל צילום.

המערכת למדה לזהות מאפייני כביש ואת התגובה הרצויה בהתאם למיקום הרכב ותנאי הדרך. לסיכום, גישת end-to-end עבור בעיית הרכב האוטונומי השיגה תוצאות מעולות.

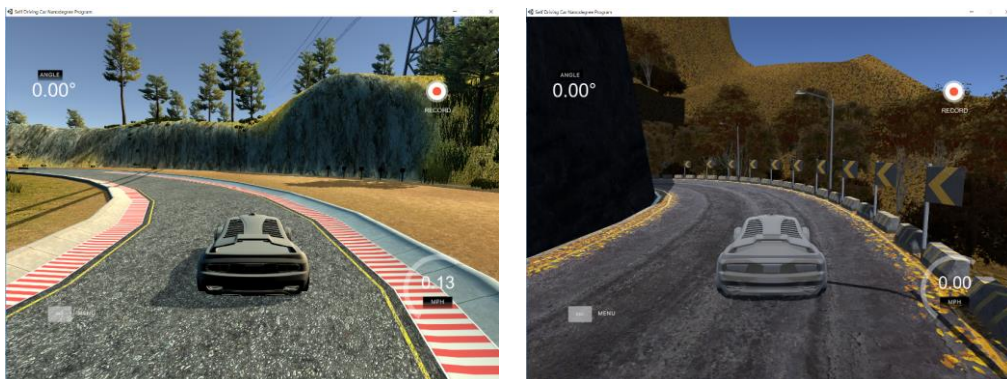


תיאור המערכת

בפרויקט זה פיתחנו מערכת End-To-End מבוססת רשת CNN לנהיגה אוטונומית בסימולטור. הרשת לומדת לחקות את נהיגת האדם על סמך הקלטות נהיגה.

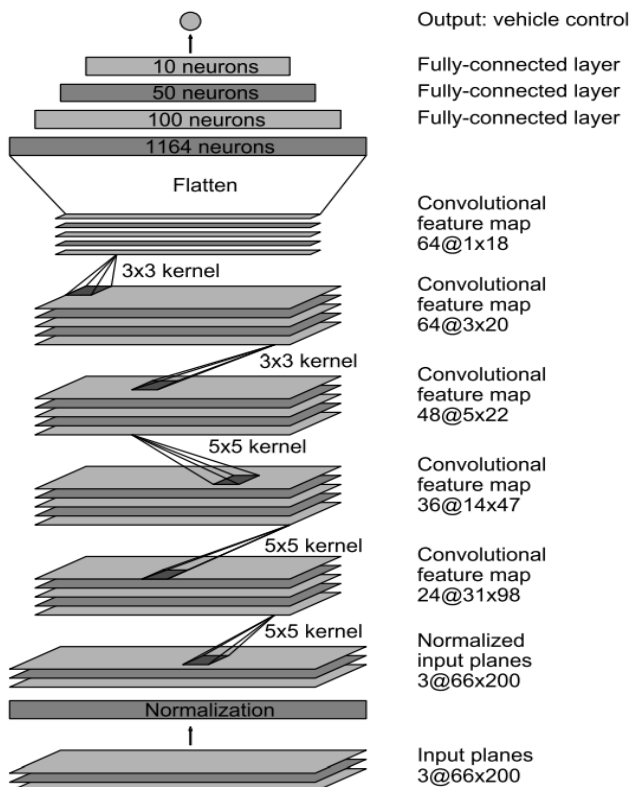
הסימולטור

נעזרנו בסימולטור קוד פתוח שנכתב ב-Unity. הוא משמש את Udacity לקורסים שעוסקים בלמידה עמוקה. הסימולטור מכיל 2 מסלולים, ותומך בהקלטת נתוני נהיגה (תמונות מ-3 מצלמות בקדמת הרכב, מהירות, זווית היגוי, דוושת גז ודוושת מעצור).



הרשת

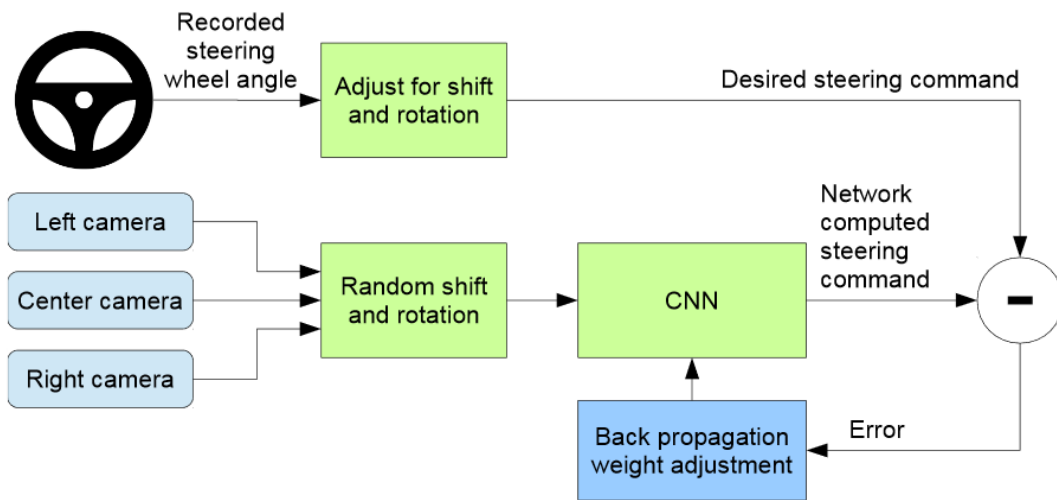
מבוססת על הרשת במאמר של Nvidia, עם מספר שינויים:



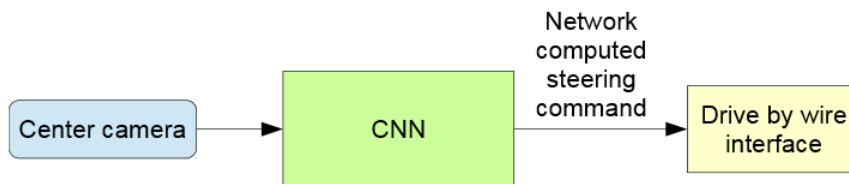
- הורדת מימד – במאמר, NVIDIA השתמשו ב-strided convolution בשכבות הראשונות כדי להקטין את המימדים של כל תוצאה. אנחנו מצאנו ש-max-pooling בין שכבות הקונבולוציה השיג ביצועים טובים גם כן, ולכן בחרנו להשתמש בו. לעומת זאת, בבחינות מודלים רבים ראינו ששימוש ב-avg-pooling גרם לרכב לנהוג באופן פחות מספק ויזואלית.
- אקטיבציה – במאמר, NVIDIA השתמשו בפונקציית ELU בכל השכבות. ניסינו לשנות את פונקציית האקטיבציה בשכבות ה-FC ל-tanh כדי לשמור על איזון בין זוויות חיוביות לשליליות, אף על פי שישנן טענות שלא אמורה להיות לכך השפעה כי הרשת יכולה ללמוד להתגבר על חוסר איזון זה. לבסוף בחרנו להישאר עם מבנה של אקטיבציה ELU בשכבות הקונבולוציה ו-tanh ב-FC מכיוון שזה נתן ביצועים טובים בבחינות מודלים רבים.



בזמן אימון הרשת מקבלת נתונים מ-3 המצלמות ואת זווית ההיגוי המתאימה.

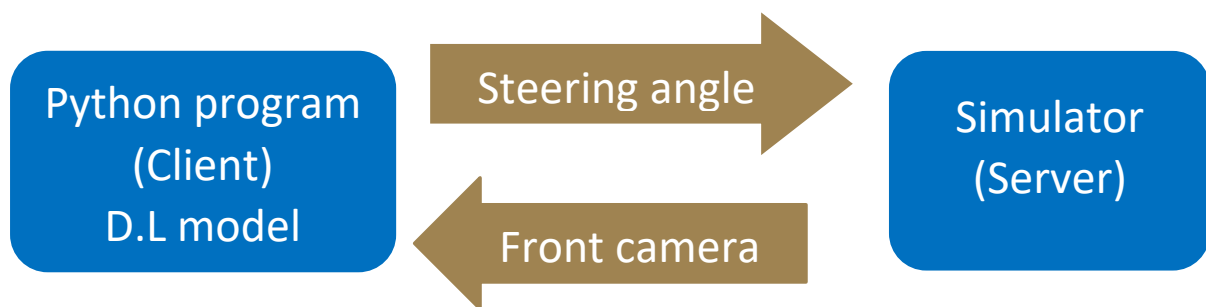


הרשת המאומנת מייצרת פקודת היגוי בהתאם לקלט ממצלמה בודדת (האמצעית).



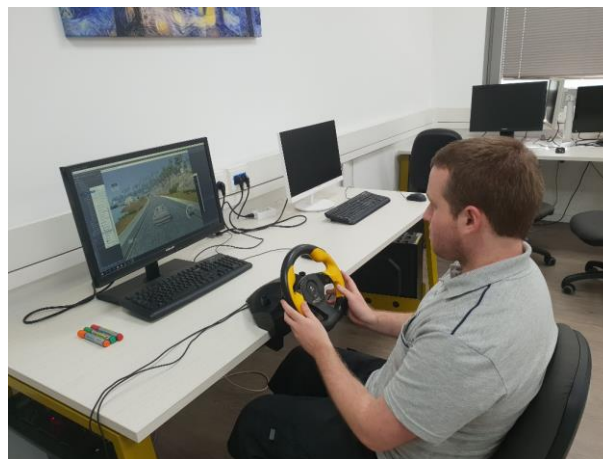
הפעלת הרשת ותקשורת עם הסימולטור

המודל מופעל על ידי תוכנת פייתון. התקשורת עם הסימולטור מתבצעת בעזרת סוקטים. הסימולטור מוציא כפלט תמונה מהמצלמה המרכזית, ומקבל פקודת היגוי. הרשת יעילה ופועלת בזמן אמת.



איסוף נתונים

הקלטנו נהיגה באחד המסלולים של הסימולטור. ההקלטה מכילה אוספי מידע רגעיים. כל אוסף רגעי מכיל 3 תמונות ממצלמות בקדמת הרכב (ימין, מרכז, שמאל), מהירות, תאוצה, תאוצה חזוית היגוי. הקפדנו על נהיגה נורמטיבית, בהתאם למתואר במאמר. ההקלטות נשמרו בקבצי CSV.



	A	B	C	D	E	F	G	H	I
1	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.118857	0	24.99913		
2	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.118857	0	24.89332		
3	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.118857	0	24.76758		
4	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.118857	0	24.66351		
5	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.118857	0	24.56033		
6	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.118857	0	24.45803		
7	C:\Users\r	C:\Users\r	C:\Users\r	-0.10917	0.118857	0	24.35303		
8	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	24.23151		
9	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	24.14024		
10	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	24.05186		
11	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	23.96494		
12	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	23.8792		
13	C:\Users\r	C:\Users\r	C:\Users\r	-0.12854	0.12854	0	23.77768		
14	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.12854	0	23.69523		
15	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.12854	0	23.61442		
16	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.12854	0	23.53378		
17	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.12854	0	23.45379		
18	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.002663	0	23.37439		
19	C:\Users\r	C:\Users\r	C:\Users\r	-0.04139	0.293149	0	23.35613		

כל שורה בקובץ ההקלטה מכילה מידע מפריים אחד. השדות משמאל לימין:

- מיקום (בדיסק) של תמונה ממצלמה אמצעית
- מיקום (בדיסק) של תמונה ממצלמה שמאלית
- מיקום (בדיסק) של תמונה ממצלמה ימנית
- זווית היגוי
- דושת תאוצה
- דושת תאוצה
- מהירות

עיבוד נתונים

המערכת שמתוארת במאמר מיועדת לרכבים אמיתיים. איסוף המידע התבצע בכבישים בהם יש כלי רכב אחרים והולכי רגל לכן בזמן איסוף המידע צריך לבצע נהיגה נורמטיבית וחוקית – שמירה על מרכז הנתיב, פניות מתונות. רצינו לממש בסימולציה את המערכת המתוארת במאמר ולכן ההקלטות שלנו בסימולציה בוצעו באותו אופן נהיגה.

נתקלנו במספר אתגרים:

- חלק מאיזורי התמונה לא רלוונטים להיגוי
- זוויות ההיגוי בעלות שונות קטנה עם תוחלת שקרובה ל-0. בנוסף, אין זוויות קיצוניות. בהינתן מידע כזה המערכת לא תוכל ללמוד איך להתמודד עם פניות חדות.
- הרשת צריכה ללמוד איך לתקן טעויות נהיגה כמו סטייה ממרכז הנתיב. מאחר וההקלטות בוצעו בנהיגה נורמטיבית, אין מידע כזה במאגר הנתונים.
- המידע נאסף במסלול אחד שהוא ברובו מישורי וחסר צללים. חלק מהבדיקה בוצעה במסלול שהמודל לא ראה באימון, ובו יש פניות חדות, שינויי גובה וצללים רבים. בתחילת הפרויקט המודל התקשה עם המסלול השני.

כדי להתמודד עם אתגרים אלה ביצענו עיבוד של הנתונים והצלחנו לשפר משמעותית את הביצועים.

חיתוך תמונה

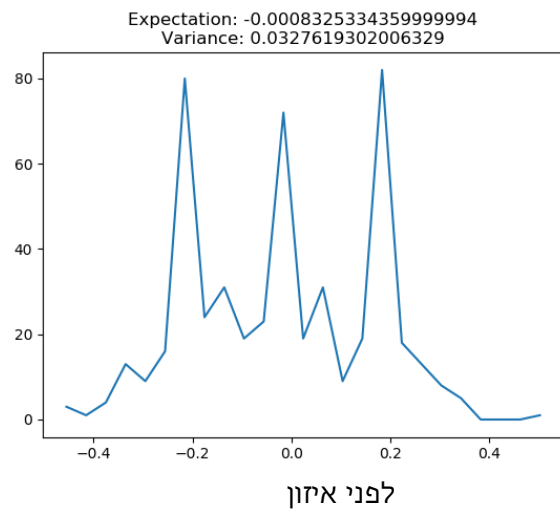
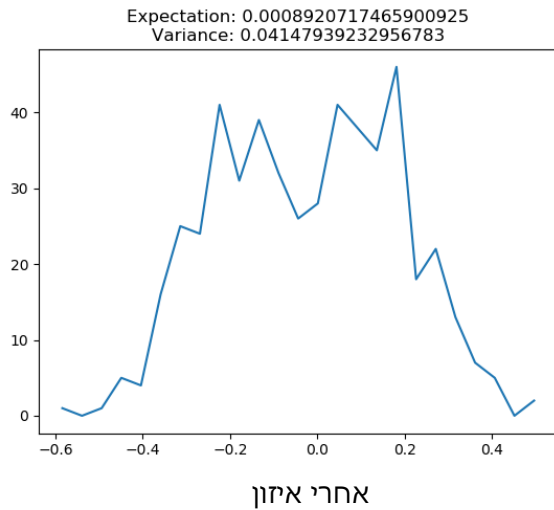
חלק מהנתונים בתמונה לא רלוונטים להיגוי – לדוגמא שמיים ומה שנמצא הרבה מעל קו האופק. מידע זה ישפיע לרעה על זמן וביצועי הלמידה. לכן ביצענו חיתוך לתמונה.





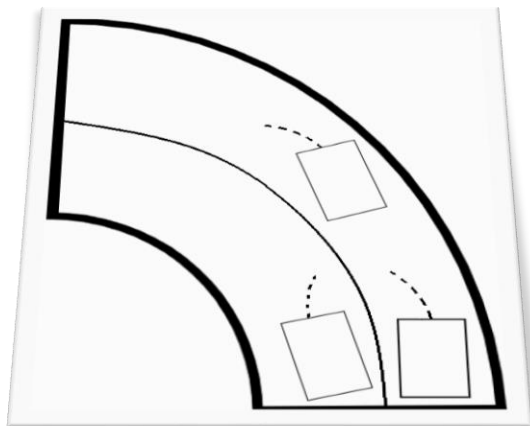
איזון זוויות היגוי

הגדלנו את הסיכוי של זוויות גבוהות להיכנס לאימון. בנוסף ביצענו שיקוף התמונה והפיכת זווית ההיגוי. באופן זה השונות של זוויות ההיגוי גדלה והתוחלת התקרבה לאפס. בנוסף נמנע bias לזוויות קטנות ולצד מסוים ונוצרו דוגמאות נוספות לאימון.

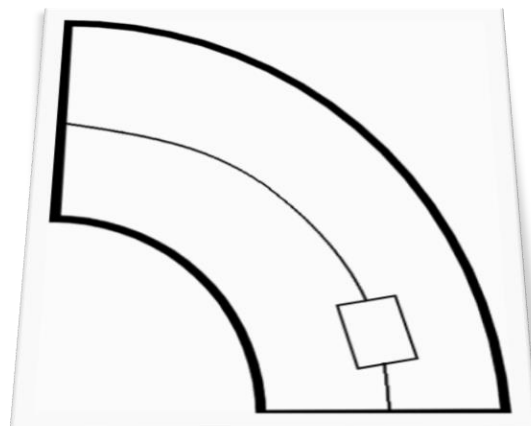


יצירת תרחישי נהיגה נוספים

הרשת צריכה ללמוד כיצד לתקן טעויות היגוי כמו סטייה מהנתיב. מאחר וההקלטות בוצעו בנהיגה נורמטיבית, אין מידע כזה במאגר הנתונים. לכן הוספנו תמונות בהן הרכב לא נמצא במרכז הנתיב (שינוי מיקום) וכאשר הרכב לא מקביל לשולי הכביש (שינוי זווית).



תרחיש מסומלץ



תרחיש מוקלט

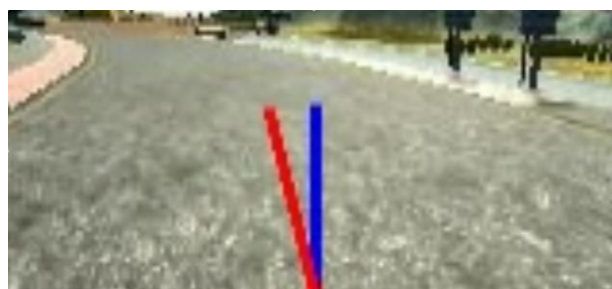
יצרנו תרחישים בהם הרכב לא ממוקם במרכז הנתיב בעזרת 2 המצלמות הצדדיות. בנוסף ביצענו Affine transformation כדי לסמלץ זוויות שונות אליהן הרכב פונה. בתהליך זה נגרם עיוות וירידה באיכות התמונה, אבל למרות זאת זה שיפר מאוד את ביצועי הרשת. לצורך המחשה מצורפת תמונה ממאגר הנתונים ושתי הזוויות שלה. באדום זווית ההיגוי המתאימה.



תמונה מקורית



סיבוב התמונה שמאלה



סיבוב התמונה ימינה



אוגמנטציה

לצורך שיפור הלמידה של המודל, ביצענו שינוי בהירות והוספת צללים. במסלול שעליו לא הקלטנו מידע לאימון כלל יש צללים רבים ובהירות נמוכה, לכן אוגמנטציה מסוג זה עזרה לטסטים במסלול זה.





תהליך האימון

כשמאמנים מודל ללמידת משימה, קיימים פרמטרים רבים שניתן ללמוד תוך כדי אימון, או לקבוע אותם בתור *hyper – parameters* שאותם ניתן לבחון ולכוון, תוך כדי אימון מודלים רבים.

הפרמטרים שקבענו הם:

פרמטר	טווח	גודל מרווח	סה"כ אפשרויות	משמעות
Small_angle_keep_probability	[0-0.8]	0.1	8	הסתברות להשתמש בזווית היגוי נמוכה לאימון
Augment_probability	[0.2 – 0.8]	0.1	6	הסתברות לבצע אוגמנטציה למידע
Translate multiplier	[0.001 – 0.005]	0.001	4	מכפיל ביצירת תרחישים נוספים
Batch size	[10 – 500]	20	25	גודל batch
Activation functions	elu, relu, tanh		3	אקטיבציה לשכבות FC
Pooling layers types	avg, max		2	טכניקת הורדת מימד בין שכבות הקונבולוציה

מחישוב של כלל האפשרויות למודלים בשיטה זו נקבל שמספר המודלים שיש לבחון הוא:

$$8 \cdot 6 \cdot 4 \cdot 25 \cdot 3 \cdot 2 = 28800 \text{ models}$$

זה מספר גדול מידי של מודל ולכן תהליך הסינון התקדם בשלבים, כל בכל שלב אימנו כמה מאות מודלים, הסקנו סטטיסטית אילו פרמטרים תורמים יותר לנהיגה טובה ואיתם המשכנו בניסויים הבאים.



בהינתן הפרמטרים הנ"ל, קבענו את האלגוריתם לבחירת דגימות אימון באופן הבא:

```
def generate_batch(x_train, y_train, batch_size, augment_prob, small_angle_keep_prob, translate_multiplier):  
    x_train, y_train = permutation(x_train, y_train) # random permutation of the data  
    perm_i = 0  
    batch_array = empty[batch_size] # allocate an empty array with the size of the data and batch_size  
    while True: # generating more batches while the loop doesn't break  
        batch_i = 0  
        while batch_i < batch_size:  
            if random(0,1) < augment_prob:  
                image, steer = augment(x_train[perm_i], y_train[perm_i], translate_multiplier)  
            else:  
                image, steer = load(_train[perm_i], y_train[perm_i])  
            if abs(steer) < steer_threshold and random(0,1) > small_angle_keep_prob:  
                continue  
            batch_array.append(image, steer) # append the new data to the batch  
            batch_i += 1  
            perm_i += 1  
            perm_i %= length(x_train)  
        yield batch_array
```

תהליך ייצור *batch* לאימון מודגש באפור באלגוריתם. ניתן לראות לפי הפסאודו קוד שהאלגוריתם מייצר לפי השלבים הבאים:

1. מבצע אוגמנטציה למידע לפי הסתברות מסויימת
2. מתעלם מזווית קטנה בהסתברות מסויימת
3. אם לא הגענו לגודל מספיק של *batch*, חוזרים לתחילת מערך המידע לאימון (ע"י חישוב מודולו) וממשיכים.

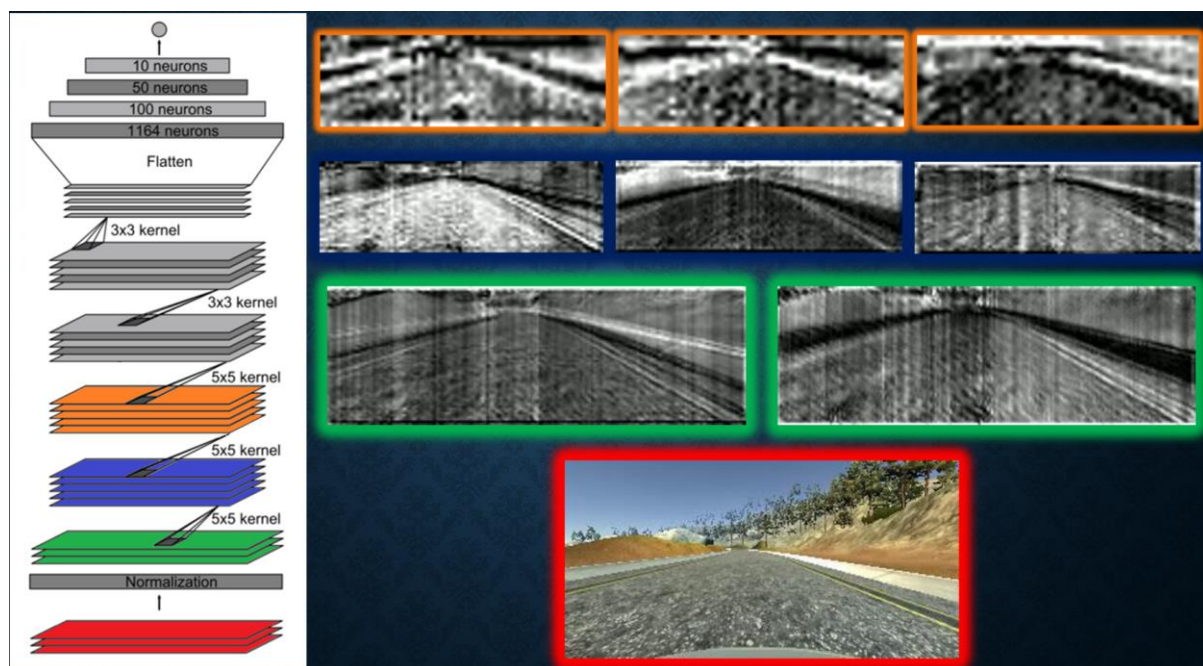
TUNING HYPER PARAMETERS

לאחר בחינת המודלים, הגענו לכיוון הבא:

הערך שנבחר	השערה מדוע סיפק תוצאות טובות	פרמטר
0.2	הגדלת ערך השונות של זוויות היגוי	Small_angle_keep_probability
0.5	כנ"ל	Augment_probability
0.003	כנ"ל	Translate multiplier
80	ייתכן שההסבר לכך הוא ש-batch size קטן גורם לעדכוני משקלים יותר סטוכסטיים ולכך יש השפעה חיובית על תוצאות האימון – זה יכול לגרום לרשת לצאת ממנימום מקומי ולמצוא אזור בפונקציה שהוא יותר שטוח והרבה נקודות בסביבה מקבלות ערך שקרוב למינימום	Batch size
tanh	אף על פי שרשת יכולה ללמוד להתגבר על סטייה של פונקציות לערכים מסוימים (elu למשל נוטה לערכים חיוביים), בחרנו בפונקציה האי-זוגית tanh כי ראינו יותר חופשיות בהיגוי	Activation functions
max	הוספת יכולת נוספת לרשת ללמוד התנהגות לא-לינארית	Pooling layers types

FEATURE MAPS

ניתן להלן הפלט של השכבות הראשונות של הרשת על קלט נתון:



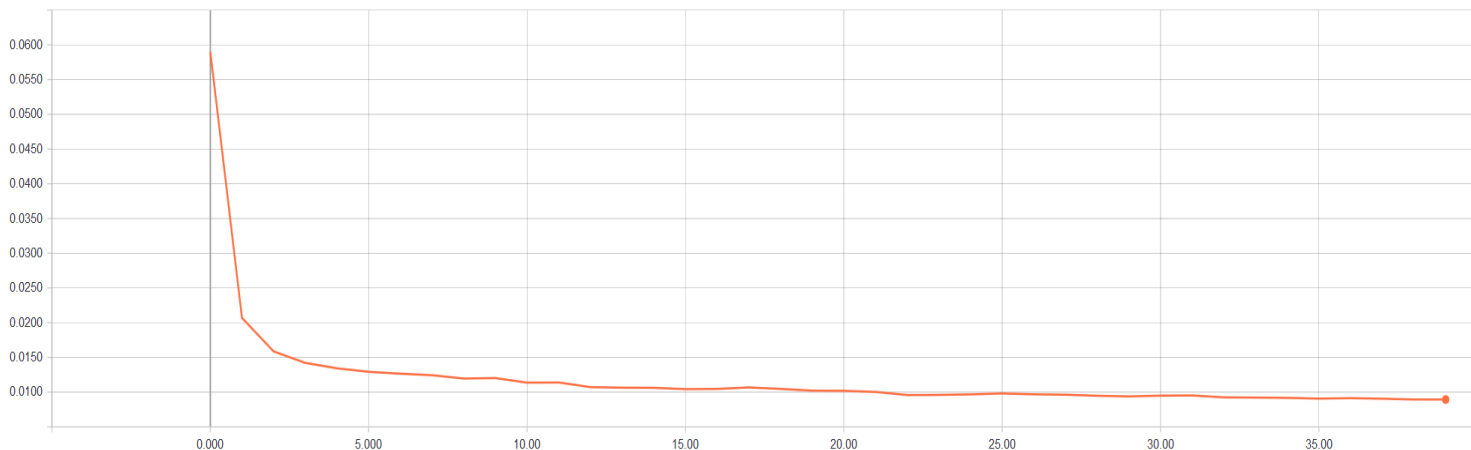


ניתן לראות בבירור ששכבות הקונבולוציה הפרידו את שולי הכביש, ואנו מסיקים מכך שזה פרמטר שהרשת קבעה שמשמעותי עבור בחירת זווית היגוי.

הערה – לא הוספנו תמונות מעבר לשכבה השלישית כי מימדיהן קטן ולא ניתן היה להסיק את משמעותן.

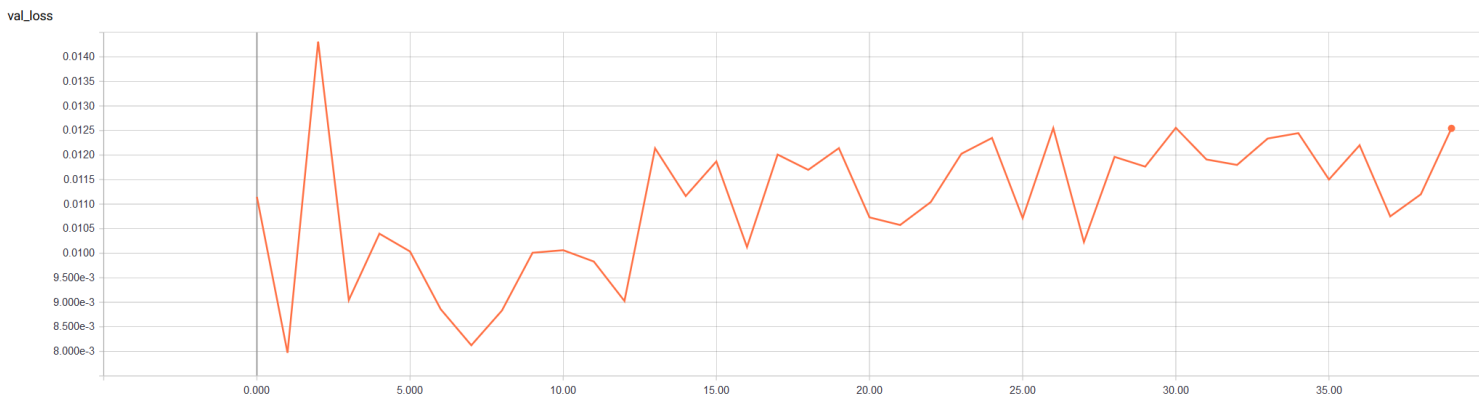
זמני אימון

טווח הזמנים לכל epoch נע בין חצי דקה לדקה. כתלות בגודל ה-batch. להלן גרף ה-loss כפונקציה של מספר ה-epochs:



לפי הגרף ניתן לראות שלאחר כ-10 אפוקים, קיבלנו התכנסות ולמעשה יכולנו להפסיק את תהליך האימון עם תוצאות מספקות.

בנוסף, בזמן בחינת המודלים, גילינו תופעה שבה מודלים שאומנו מעל 20 epochs נכשלו בפניות יחסית קשות (זוויות חדות בעיקר), אף על פי שיש loss יותר נמוך על ה-training set. הסבר לכך ניתן לקבל ע"י הגרף הבא, המתאר את ה-loss על ה-validation-set:



ניתן לראות בבירור עלייה של ה-loss ב-epochs מאוחרים. יכולים להיות לכך מספר סיבות:

- ה-learning rate בשלב זה גבוה מידי ולכן אנו יוצאים ממינימום שהיינו בו יותר מוקדם
- חוסר יכולת של הרשת להכליל – פתרון אפשרי הוא להוסיף רגולריזציה לשכבות ה-FC ולהגדיל את ה-dropout.



מסקנות

המטרה שלנו הייתה ליצור מערכת מבוססת deep learning בתחום נהיגה אוטונומית. להלן המסקנות העיקריות שלנו מהפרויקט:

- רשת קונבולוציה יכולה ללמוד בהצלחה את משימת ההיגוי לפי תמונה בלבד.
 - מספיק זמן אימון קצר כדי להשיג תוצאות מספקות (כרבע שעה).
 - לעיבוד מקדים של הנתונים יש חשיבות רבה – ללא עיבוד מקדים, המידע לא היה מאוזן ולכן המודל לא הכיר מצבים שעלול להיתקל בהם. העיבוד המקדים היה הכרחי כדי לעזור לרשת ללמוד לשמור על הנתיב כאשר יש סטיות ממנו.
 - כיוון ה-Hyper-Parameters עדין וכל שינוי משפיע על תוצאות המודל. לדוגמה, הוספת כמות לא מספקת של צללים לתמונה גרם לחוסר יכולת של הרשת להתמודד במצבים חשוכים.
- לסיכום, נהנינו מאוד מהפרויקט שלפני כמה חודשים נראה לנו כמו מדע בדיוני.

המלצות

יש תועלת גדולה לסימולציות בעולם ה-DL בכלל, ובתחום הרכב האוטונומי בפרט:

- בטיחות - ניתן לאמן ולבדוק מערכות מורכבות בסביבה בטוחה
- מחיר - הרצת רכב בסימולציה זולה יותר
- בדיקותיות - בסימולציה ניתן לייצר ולשחזר מקרי קצה. ניתן לבדוק מקרים שההסתברות להיתקל בהם בעולם האמיתי אפסית, וכך להגיע לרמת בדיקותיות גבוהה יותר.

במהלך הפרויקט כדי לבדוק את ביצועי המערכת, צפינו באופן הנהיגה של המודל. זו הייתה משימה ממושכת. זה העלה לנו רעיונות לפיצ'רים רצויים עבור סימולטורים:

- חשיפת נתונים על התנגשויות
- אפשרות להגדיר נתיב נסיעה רצוי (מרכז נתיב, פנייה אידיאלית...) וקבלת מידע על מיקום הרכב ביחס לנתיב הרצוי
- תמיכה בהרצה אוטומטית של הסימולציה עם מודלים שונים

פיתוח כלי סימולציה התומכים בפיצ'רים כאלה יוכלו לתרום למחקר ופיתוח מודלים חדשים.

תודות

אנו רוצים להודות לרון סלוסברג, ירון חונן ולמעבדת GIP על העזרה והמשאבים שהושקעו בנו במהלך ביצוע הפרויקט.