



**CRMIL**

Control Robotics and Machine Learning Laboratory

**הטכניון - מכון טכנולוגי לישראל**

**TECHNION - ISRAEL INSTITUTE OF TECHNOLOGY**

**הפקולטה להנדסת חשמל  
המעבדה לבקרה רובוטית ולמידה חישובית**

**דו"ח פרויקט : פרויקט א'**

# AI Algorithm Training

**מגישיים:**

Yonathan Bettan

**מנחה:**

Eden Sasson

**סמסטר: חורף**

**שנה: 2019**

**תוכן עניינים:**

19 .....	1.	מטרת הפROYיקט
20 .....	2.	מבוא ורקע תיאורטי
22 .....	3.	תהליך המהקר והדרך שעברנו
24 .....	4.	תיאור חומרה
28 .....	5.	תיאור תוכנה
29 .....	5.1	מבנה התוכנה
30 .....	5.2	הסבר על התוכנה
32 .....	6.	תוצאות ומסקנות
34 .....	7.	נספחים
35 .....	7.1	הוראות הפעלה
36 .....	7.2	מאמרם
37 .....	7.3	מקורות

## **1. מטרת הפרויקט:**

הנήג אוטונומית התפתחה מעבר למצופה בחצי עשור האחרון, מחלומות מטאורפים לפרויקטים ממשיים. הנήג אוטונומית נהייתה אחת מהטכנולוגיות הגדולות וה蛮תוחות ביותר לעיצוב פני העתיד. מחשבה כי בעוד מספר שנים נוכל למצוא רכבים ללא הנגיג משטייטים להן על הכבישים הינה מציאותית ביותר.

את התפתחות תחום הנήג האוטונומית ניתן לגוזר מהתפתחויות במספר רב של תחומי, כגון:

- תוכנה (בינה מלאכותית)
- חומרה (כח חישובי גבוה)
- חישובים מבזוריים (ענן).

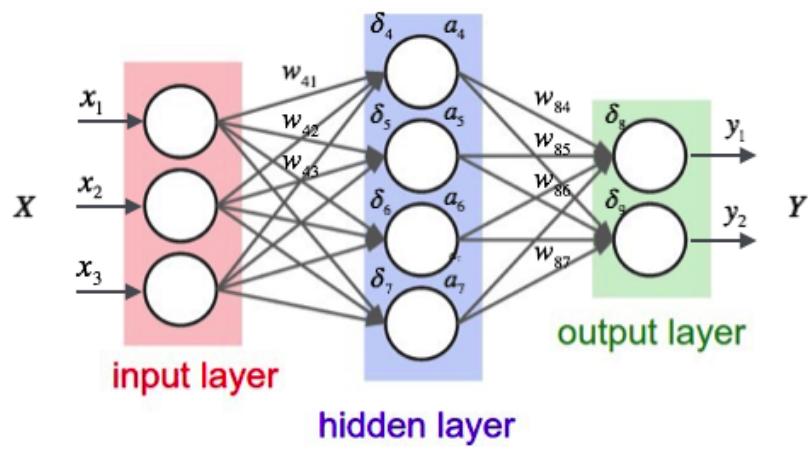
התפתחויות אלו אפשרו לנו לנצל ולעבד מידע רב, דבר המאפשר לנו למתוח את הגבול הידע עד כה.

למרות היכולת לעבוד מידע רב, המידע הנאגר יום יום אין בו די בכך לספק את הדרישות של מערכות נבונות מתקדמות. דרך אחת בכך ליצור מלאכותי בתיחסים שונים הינה על ידי סימולטור. סימולטור מקל לא רק על איסוף מידע מגוון, אשר היה לו קח ימים אם לא חודשים בפועל מציאותי, אבל בבד סימולטור מספק סביבת עבודה זולה, ומהירה.

ישנו מספר רב של מודלים וטכניקות ללמידה סוכנים בסימולטורים. מטרת הפרויקט הינה לספק שלד אלגוריתמי לצורך הנגשת אימון סוכנים עבור רכבים אוטונומיים ובפרט עבור פרויקט הפורמללה האוטונומית.

## 2. מבוא ורקע תיאורטי:

או בעברית רשת נוירונים, הינה מודל שטחני לחקר את מבנה המוח האנושי המורכב מנוירונים רבים המתקשרים ביניהם על מנת לעבוד מידע. הרשת מיוצגת ע"י גרף ממושקל המורכב מ- input nodes ו- hidden nodes ,nodes המכילים תוצאה כלשהי בהתאם לפונקציה על ה- output nodes ומשקלוי הקשיות שלהם כפי שמתואר באיוור 1.#.



איוור 1

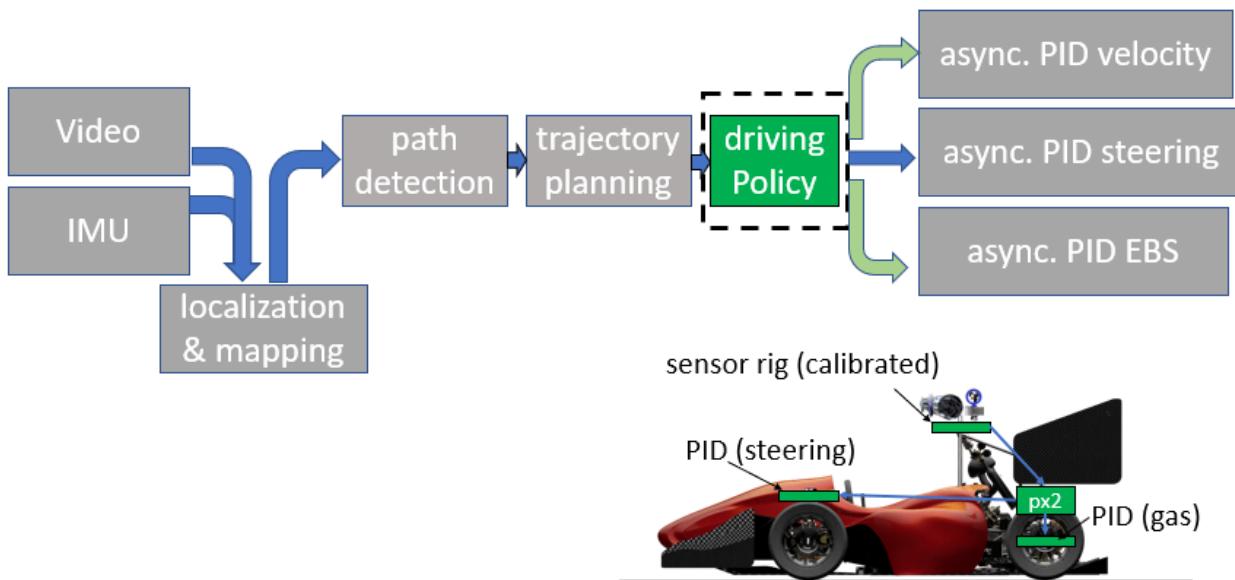
הרשת לומדת מחזוקים שהוא מקבלת, בשלב ראשון נתונים לרשת קלט כלשהו והוא מניבת עבورو פלט, שאינו נכון כלל מן הסתם, לאחר מכן חיזוק חיובי או שלילי לרשת על מנת הדיק שלה בתוצאה, הפער בין התוצאה הרצוייה למציאות עבר תהליך שנקרא backward propagation כך שבעצם הسطיטה "מפעעתה" אחורנית ל- hidden nodes והרשות מעכנת את משקלוי הקשיות שלה על מנת לקבל את התוצאה הרצוייה. חזורים על תהליך זה מספר רב של פעמים עד שהרשות מגיע לדיקוק מספק וזה בעצם תהליכי הלמידה.

כל שמתבצעות יותר איטרציות ונ נתונים לרשת יותר דוגמאות כך היא נהיית "חכמה" יותר ומדויקת יותר.

ארכיטקטורה:

מבט כללי:

#1 איור #1



#2 איור #2

איור #2 מתראר את התכנון הטיפוסי לפתרון בעיות מהסוג הנ"ל:

1. קבלת מידע אודוות הסביבה באמצעות החישונים והמצאות במרחב
2. תכנון מסלול נסיעה עתידי
3. המרת המסלול העתידי לפקודות היגוי

קלט: תמונות, ו-IMU - מד אינרצייה.

שכבה נסתרת 1: חילוץ וקטור התקדמות.

פלט: גז והיגוי לשיליטה ובקרה על הרכב.

### 3. הדרך שעברנו:

ישנם שלושה שלבים עיקריים באימון מערכת נבונה:

1. איסוף מידע
2. עיבוד המידע
3. אימון מודל
4. בדיקת מודל

#### איסוף מידע:

בכדי לאמן סוכן שישלוט ברכב, علينا לאסוף מידע על התנהגות הרכב במצבים שונים. בזמן פניה, או התחממות ממושול, האצה או האטה.

השתמשנו בספר מתודות בכדי לאסוף מידע על הרכב:

a. איסוף מידע ידני – ביצענו מספר נסיעות ברכב, בהן שלטנו ברכב על ידי המקלדת. הקלטנו את

המידע ובחנו את צורתו המידע אשר הקלטנו.

יתרנו שיטה זאת, איסוף מדויק של המידע – שליטה במיקום הרכב ובהתנהגות הרכב.

חסרונו שיטה זאת, בכדי לאסוף מידע בצורה זאת נדרש כי אדם ינהג באופן עצמאי ברכב, דבר זה דורש המון שעות נהיגה.

חסרונו נוסף כי על ידי פקודות המגיעות מן המקלדת ניתן לבצע רק פקודות קיזזניות. פניה ימינה או שמאליה במקלדת שקולות לסייע ההגה עד לקצה הקיזוני במכה, ללא יכולת שליטה עдинה. כמו כן עבור הגז והברקס.

b. איסוף מידע אוטומטי – כתבנו מספר סקריפטים אשר שלטו ברכב וננתנו לו פקודות בקרה

רנדומליות. אופן תהליך זה:

כל 0.3 שניות חוט התעוורר, שאב מהסימולטור מידע IMU ותמונה קדמית של הרכב.

חות זה קבע את היגוי הרכב להיות X ואת תאוצת הרכב להיות Y כאשר X ו-Y נלקחים מתחם התפלגות נורמללית על פי אופי המידע החסר.

עבור דגימות של נסיעה ישירה, תוכלת 0 ושונות נמוכה, עבור דגימות של פניה ימינה, תוכלת 0.5 ושונות ביןונית, וכן עבור פניה שמאליה עם תוכלת -0.5.

הרכב חוזר על התהליך עד להתגשות, בעת התגשות אנו מתחילה את הסימולטור ומתחילה נסעה חדשה.

יתרונות שיטה זאת, ניתן לאסוף המון מידע, ניתן לאסוף מידע על פועלות עדינות. חסרונות שיטה זאת, מירב הנסיעות היו קצורות עקב התגשות בעצמים, המידע מתאר פועלות רנדומלית על התגשות הרכב אשר אין בהכרח מייצגות את הפעולה אותה אנו מחפשים. c. איסוף מידע מהאינטרנט – אספנו מספר קבצי מידע ממוקורות שונות אשר השתמשו בסימולטור.

יתרון שיטה זאת – קיבלנו מידע טוב ללא מאץ. חסרון שיטה זאת – יש לנו אפס שליטה בתוכן המידע ובאופי הנסיעות. אך כן ראיינו והשנו את תוכן הקבצים.

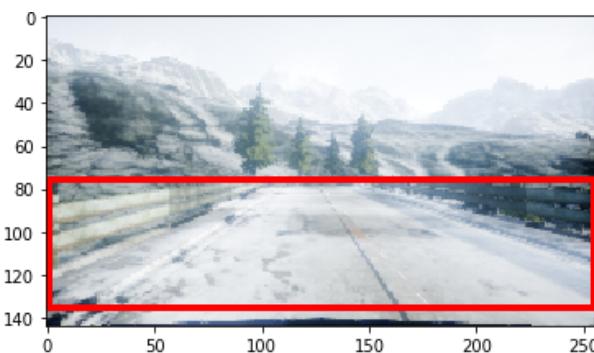
### עיבוד המידע:

בכדי לאמן מודל על בסיס מידע קיים, נרצה "לחוש" את המידע, כלומר לראות איזה סוג מידע קיים, ואילו מניפולציות אלו יכולים לבצע על המידע בכדי לקבל מידע יעיל וחכם יותר. נתאר מספר מניפולציות כאלו:

1. **מניפולציות על תמונות שנאספו –** רשותנו נוירוניים הינהן רגישות לשינויים בתמונות אותן הם מקבלות, דבר זה יכול לפגוע קשה בטיב רשות אשר פועלת בתנאים שונים מאשר התנאים בה היא למדה. בכדי להקטין את ההשפעות הללו ביצענו את הפעולות הבאות:

a. הקטינו את מסגרת התמונה כך שתכלול בעיקר את הכביש, ככלומר הורדנו את החלק העליון של

התמונה כך שלא תכלול נוף כפי שניתן לראות באירור #3



איור #3

b. שיחקנו עם בהירות התמונה, בכך אנו מייצרים תמונות בעלי תוחלות צבעים שונות, דבר אשר השפיע רבות על הרשות.

c. ביצענו תמונות מראה לתמונות – דבר זה מכפיל את כמות הדגימות ללא מאץ רב.

2. **מניפולציות על מידע ה-IMU אשר הקלטנו:**

a. בזמן נחיתה בסימולטור אנו יכולים לאסוף מידע על מיקום הרכיב. על ידי סיפוק לרשות מספר רב של נקודות ומידע על הרכיב בנקודות אלו אנו יכולים לגוזר מידע רב כגון מהירות צדדים וקוויים, תאצוזות, וקטור התקדמות.

ב. מאחר וראינו כי וקטור ההתקדמות הינו חשוב ביותר, ביצענו טרנספורמציה לוקטורים אלו ממיישר קרטזי למישר פולרי ובכך "הקלנו" על הרשות בבחירה זווית ההיגוי הבאה והקטנו את מידת הקלט.

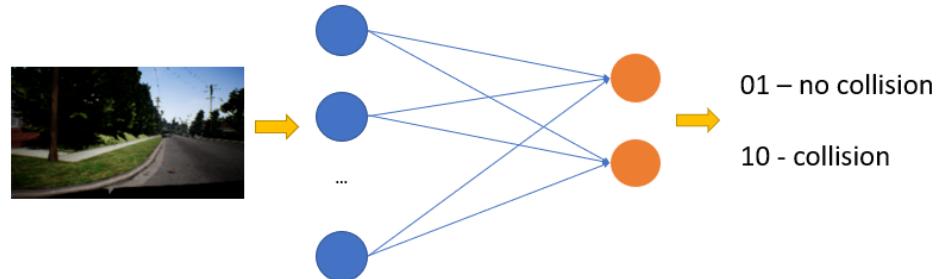
שלב נוסף בעיבוד המידע הינו ניתוח כמות וסוג המידע. לדוגמה בניסיונות ידניות מירב הדגימות הינה של מצבים בהן הרכב נושא ישיר. בזמן אימון רשות על בסיס מידע כנ"ל, מסווג בוורח למצע את ההיגוי והתואזה ל-0. דבר זה אינו שגוי, על פי בחירות אלו הרשות "כולעת" בצורה טובה על סט האימון, נשיאה ישירה. ניתן להגדיר מצב זה Under Fitting. סט המידע אינו מספק די מידע על המציגות. בכך לחסוך מצבים כאלה כתבנו סקריפטים אשר מבצעות נסיונות בהן היגוי ותואצת הרכב נלקח מトー התפלגות נורמללית אשר סיפקה את ההתנגדות החסורה במידע.(Clomer, אם חסר מידע על פניות שמאליה תוחלת ההיגוי היה קטן מ-0 והשונות בהתאם).

#### אימון ובדיקה מודלים:

בכך לאמן מודל מוצלח החלטנו לבנות שכבות שונות ולראות את השפעתם לאורך התקדמות הפROYיקט. מתווך כך מצאנו קשר חזק בין אימון המודל ובחינתו על בסיס נסיונות מבחן. כך למשל, בשלב הראשון אימנו מודל פשוט על בסיס מספר קטן של דוגמאות הידוע לעצור את הרכב טרם התנגדות עצם זר סטיי כפי שניתן לראות באIOR #4. את המבחן מדכנו על בסיס יכולת הסוכן למנוע התנגדות.

Collisions are assigned negative values

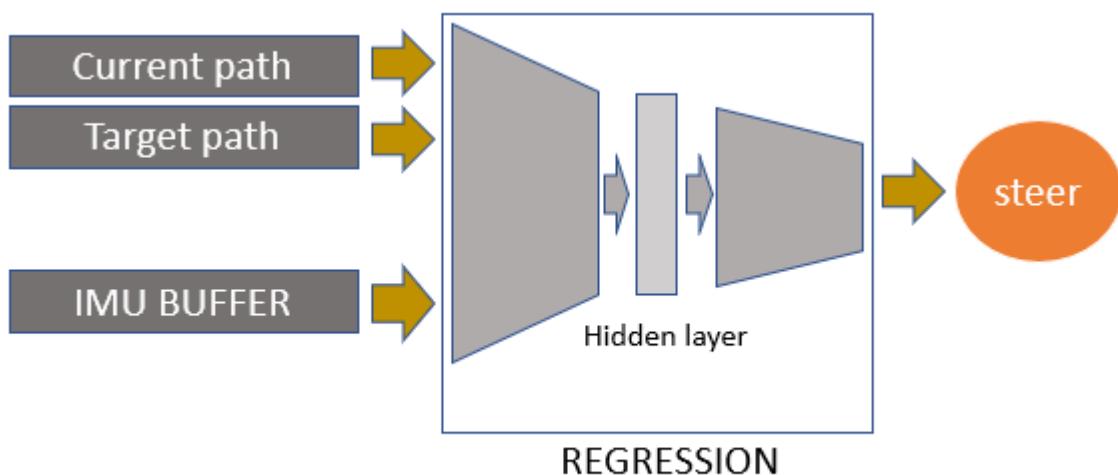
We can associate a collision event to a set of images using a Value-Function



#4 אייר

בכדי לאמן סוכן למכרים יותר סבוכים, בו עליו לבצע פניות ותאוצאות ולאו דוחק לעצור, בחרנו

להשתמש בריינסור כדי שניתן לראות באיר #5

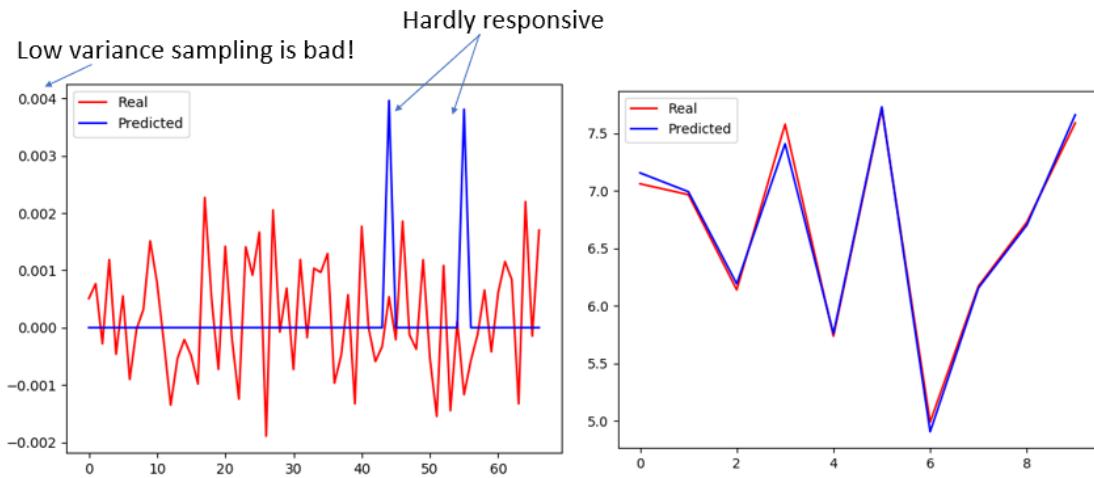


#5 אייר

עבור בעיית הרכיב האוטונומי פגשנו מספר רב של אלגורים בימוש זה, בניהם בעיה הנובעת מושנות

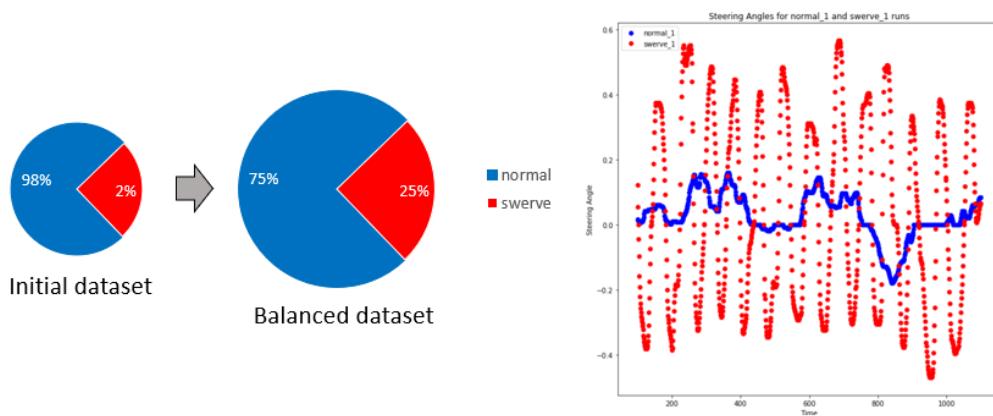
נוכחה במידע. במקרה זה ריגרסור מציע את המידע הרחוק מהתוחלת – בדוגמה נסעה "מציאותית"

הנהג נוהג את רוב הזמן בנסיעה ישירה, ריגרסור פשוט מתעלם מפניות ונוסף ישר כפי שניתן לראות באירור #6.



#6 איור

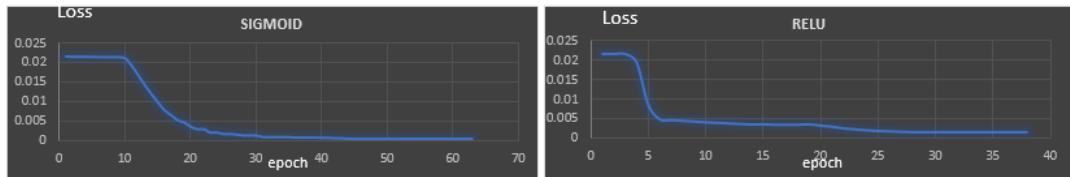
פתרון לבעה זאת הייתה לאסוף מידע בו אחוז הדוגימות בהן מתבצעות פניות הין כ-25% כפי שניתן לראות באירור #7.



#7 איור

בוחנו את יכולת המודל ללמידה ביחס לשתי פונקציות אקטיבציה שונות. קל לראות כי עבור פונקציית SIGMOID המודל לומד מהר יותר, אך מנגד לאורך זמן SIGMOID מגיע לתוצאות טובות יותר כפי שניתן לראות באירור #8

## Training Loss / Performance



#8 אירור

את מבחן והערכת הסוכנים ביצענו על בסיס תוצאות הסוכנים על הרכיב, כפרמטרים של התנגשות,  
ומרחק סטיה מהכبيש.

#### **4. תיאור חמרה:**

בפרויקט זה הרצנו את הסימולטור ואת התוכנה אשר אימהה את הסוכן על מחשב ללא שימוש ב-GPU.

#### **5. תיאור תוכנה:**

##### **5.1. מבנה התוכנה:**

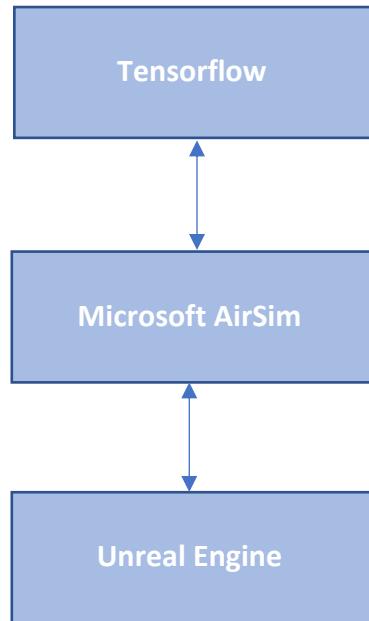
התוכנה מורכבת מ-3 חלקים עיקריים, החלק הראשון נקרא Unreal Engine והינו המנוע עצמו אשר מטרתו לפתח משחקים, סביבות עבודה שלמות כגון עיר לצורך אימון רשות אוטונומית, עיר חכמה וכו' החלק השני של התוכנה נקרא Microsoft AirSim והינו מעטפת אשר רצה מעל Unreal Engine ומספקת מודל של רכב ו- API לשיליטה על הרכב וקבלת Data מהסימולטור כגון תמונות ממצלמת הרכב, מהירות הרכב, זווית הגה וכו' החלק השלישי של התוכנה הינו כל הקשור לאימון סוכן ומתחולק לשולושה חלקים נוספים:

1. איסוף מידע ועריכתו.

2. אימון סוכן.

3. הפעלת הסוכן

איור #9 ממחיש את המבנה של התכנה שלנו.



איור #9

#### 5.2. הסבר על התכנה:

לצורך הפרויקט יש שימוש ב-3 החלקים כאשר אנו משתמשים בו- API של AirSim על מנת לשלוט על הרכיב בצורה אוטומטית ובסופו של דבר פקודות ה- API הניתן מתורגםות לפעולות על הרכיב במנוע Unreal Engine.

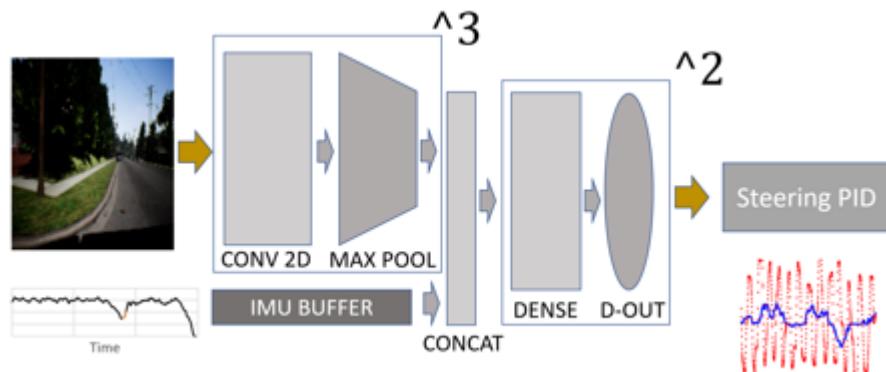
השימוש העיקרי שלנו הינו במודול של AirSim אשר מעניק לנו תקשורת server-client לשיליטה ברכיב וקבלת מידע בזמן אמת.

בחלק השלישי, עשינו שימוש בעיקר בתמונות שנתקבלו מצלמת הרכיב על מנת ללמד רשת לנגן ברכיב בצורה אוטונומית.

את מבנה המודל המפורט ניתן לראות באירועים #10-11

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 59, 255, 3)	0	
convolution0 (Conv2D)	(None, 59, 255, 16)	448	input_1[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 29, 127, 16)	0	convolution0[0][0]
convolution1 (Conv2D)	(None, 29, 127, 32)	4640	max_pooling2d_1[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 14, 63, 32)	0	convolution1[0][0]
convolution2 (Conv2D)	(None, 14, 63, 32)	9248	max_pooling2d_2[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 7, 31, 32)	0	convolution2[0][0]
flatten_1 (Flatten)	(None, 6944)	0	max_pooling2d_3[0][0]
dropout_1 (Dropout)	(None, 6944)	0	flatten_1[0][0]
input_2 (InputLayer)	(None, 4)	0	
concatenate_1 (Concatenate)	(None, 6948)	0	dropout_1[0][0] input_2[0][0]
dense0 (Dense)	(None, 64)	444736	concatenate_1[0][0]
dropout_2 (Dropout)	(None, 64)	0	dense0[0][0]
dense2 (Dense)	(None, 10)	650	dropout_2[0][0]
dropout_3 (Dropout)	(None, 10)	0	dense2[0][0]
output (Dense)	(None, 1)	11	dropout_3[0][0]
<hr/>			
Total params: 459,733			
Trainable params: 459,733			
Non-trainable params: 0			

#10 איזור



#11 איזור

### קריאה המידע מה- dataset הנתון:

קריאה פשוטה של טבלאות המידע המופיעות בפורמת csv כפי שניתן להראות באיור #12.

```
# collect the full non-images dataset
full_path_raw_folders = [os.path.join(RAW_DATA_DIR, f) for f in DATA_FOLDERS]

dataframes = []
for folder in full_path_raw_folders:
    current_dataframe = pd.read_csv(os.path.join(folder, 'airsim_rec.txt'), sep='\t')
    current_dataframe['Folder'] = folder
    dataframes.append(current_dataframe)

dataset = pd.concat(dataframes, axis=0)

print('Number of data points: {}'.format(dataset.shape[0]))

dataset.head()
```

איור #12

### עיבוד המידע:

כפי שהסביר לעיל אנו מבצעים עיבוד מידע של ה- data על מנת לספק מידע איקוטי ורלוונטי לרשות כפי  
שניתן להראות באיור #13.

```
# cooking the data
train_eval_test_split = [0.7, 0.2, 0.1]
full_path_raw_folders = [os.path.join(RAW_DATA_DIR, f) for f in DATA_FOLDERS]
Cooking.cook(full_path_raw_folders, COOKED_DATA_DIR, train_eval_test_split)
```

איור #13

## קריאה המידע המעובד לצורך אימון:

לאחר שיעבדנו את המידע ושמרנו אותו בקובץ `h5` נרצה לקרוא מידע זה על מנת להתחיל למלמת את הרשות כפי שניתן לראות באירור #14.

```
# read the coocked datasets
train_dataset = h5py.File(os.path.join(COOKED_DATA_DIR, 'train.h5'), 'r')
eval_dataset = h5py.File(os.path.join(COOKED_DATA_DIR, 'eval.h5'), 'r')
test_dataset = h5py.File(os.path.join(COOKED_DATA_DIR, 'test.h5'), 'r')

num_train_examples = train_dataset['image'].shape[0]
num_eval_examples = eval_dataset['image'].shape[0]
num_test_examples = test_dataset['image'].shape[0]

batch_size=32
```

אירור #14

## הגדרת ארכיטקטורת הרשות:

הגדרת ארכיטקטורת הרשות נעשתה (אירור #15) כפי שהסבירנו באירורים #10-11

```
# define the network architecture

image_input_shape = sample_batch_train_data[0].shape[1:]
state_input_shape = sample_batch_train_data[1].shape[1:]
activation = 'relu'

#Create the convolutional stacks
pic_input = Input(shape=image_input_shape)

img_stack = Conv2D(16, (3, 3), name="convolution0", padding='same', activation=activation)(pic_input)
img_stack = MaxPooling2D(pool_size=(2,2))(img_stack)
img_stack = Conv2D(32, (3, 3), activation=activation, padding='same', name='convolution1')(img_stack)
img_stack = MaxPooling2D(pool_size=(2, 2))(img_stack)
img_stack = Conv2D(32, (3, 3), activation=activation, padding='same', name='convolution2')(img_stack)
img_stack = MaxPooling2D(pool_size=(2, 2))(img_stack)
img_stack = Flatten()(img_stack)
img_stack = Dropout(0.2)(img_stack)

#Inject the state input
state_input = Input(shape=state_input_shape)
merged = concatenate([img_stack, state_input])

# Add a few dense layers to finish the model
merged = Dense(64, activation=activation, name='dense0')(merged)
merged = Dropout(0.2)(merged)
merged = Dense(10, activation=activation, name='dense2')(merged)
merged = Dropout(0.2)(merged)
merged = Dense(1, name='output')(merged)

adam = Nadam(lr=0.0001, beta_1=0.9, beta_2=0.999, epsilon=1e-08)
model = Model(inputs=[pic_input, state_input], outputs=merged)
model.compile(optimizer=adam, loss='mse')
```

אירור #15

### אימון הרשות:

אימון הרשות נעשה עם לכל היותר 500 epochs כפי שניתן לראות באירור #16 אך בפועל יש עצירה מוקדמת כאשר ה- loss מפסיק להשתפר.

```
# train the model
history = model.fit_generator(train_generator, steps_per_epoch=num_train_examples//batch_size, epochs=500, callbacks=callbacks,
                               validation_data=eval_generator, validation_steps=num_eval_examples//batch_size, verbose=2)
```

AIROR #16

### אתחול מצב הרכב:

בכדי לבדוק את המודל את מצב הרכב במצב בו זווית ההגה הינה 0 והרכב אינו נושא כפי שניתן

לראות באירור #17

```
# We'll set the initial state of the car, as well as some buffers used to store
# the output from the model
car_controls.steering = 0
car_controls.throttle = 0
car_controls.brake = 0

image_buf = np.zeros((1, 59, 255, 3))
state_buf = np.zeros((1,4))
```

AIROR #17

## המודל בניסעה:

ביצענו נסעה במהירות קבועה, ע"י האצה כל עוד אנחנו מתחת למהירות סף מסוימת, ונתנו לרשת לנבأ

את זווית הגהה על פי הרכב צריך לנוהג בהתאם למצבו הנוכחי והתמונה אותה הוא קולט כפי שניתן

לראות באירור #18

```
# let the model drive the car, our model don't predict speed therefore we will
# set throttle=1 when speed < 5m/s and throttle=0 otherwise.
while (True):
    car_state = client.getCarState()

    if (car_state.speed < 1):
        car_controls.throttle = 0.5
    else:
        car_controls.throttle = 0.0

    image_buf[0] = get_image()
    state_buf[0] = np.array([car_controls.steering, car_controls.throttle, car_controls.brake, car_state.speed])
    model_output = model.predict([image_buf, state_buf])
    car_controls.steering = round(0.5 * float(model_output[0][0]), 2)

    print('Sending steering = {0}, throttle = {1}'.format(car_controls.steering, car_controls.throttle))

    client.setCarControls(car_controls)
```

אירור #18

## **6. תוצאות ומסקנות:**

בפרויקט זה שמו למטרה לספק ארכיטקטורה לנויות וברית רכב אוטונומי ועמדנו בכך. בתוצאות המבחן

ראינו כי הרכיב נושא בצורה טובה בסביבה בה אומן. עם זאת, כל להבחן בשוני שיטות באות:

1. הרכיב מבצע עליה או ירידת תלולה – במקרה זה אין מידע רב בתמונות המתקבלות. חיתוך גודל התמונה

פגע ביכולת הסוכן. עם זאת אלו מקרי קצה אשר אין צפויות לקרות באטגר הFST.

2. התנהגות הרכיב על גשרים איננה צפופה – בעוד שישנם מספר דוגמאות בהן הרכיב עבר גשרים, יחד

הדוגמאות הללו נזוק מאד. בנוסף, מוצע שלו דרכו הינו מटבṭא בצורה שונה לחלוṭין על גשר או על

כביש. מתוך שני היבטים אלו התוצאה Dunn סבירה והגונית. איננו מצפים לעבר גשרים באטגר הFST

פרויקט זה בעיקר מהו בניה תשתית חזקה לפרויקט המשך, אשר יתעסק באימון רשות להיגה אוטונומית,

ולכן התוצאות מוכנות להיבטים אחרים ניתן להוסיף או לשנות בארכיטקטורה אשר אנו מספקים.

דגם של מצב הנסיעה על הרשות המאומנת ניתן לראות באיר #19-21



איור #19



איור #20



איור #21

מסקנות:

1. שימוש בארכיטקטורות מומלצות.
2. דגימת מידע שאינו טבעי, יכול לתרום רבות לאימון והתנהגות מודל בסביבה טבעית.
3. ישנה שימושה רבה לפונקציית האקטיביזציה בין שכבות. אנו מצאנו כי RELU לוידת משמעותית מהר יותר.

נקודה נוספת אף לא פחות חשובה הינה למצוא את האיזון בין מחקר אינטרנטני טרם תחילת העבודה לבין העבודה עצמה, שכן עבודת הכנה היא דבר חשוב אף במקרה הכללי רוב הלמידה נעשית תוך כדי העבודה וקשה מאוד לנביא את כל הקשיים בהם נתקל מראש שכן אנו ממליצים לבצע מחקר אינטרנטני אף לא להתמהמה ולקפוץ למיים מוקדם ככל האפשר.

## 7. נספחים:

### 7.1. הוראות הפעלה:

[מפורט ב- ybettan/FinalProject](#)

### 7.2. מאמריהם:

<http://krisbolton.com/a-practical-introduction-to-artificial-neural-networks-with-> •

נתן בעicker כרגע תיאורתי אשר יומש בפרויקט ההמשך אף חשוב להבין שכן מטרתו

של הסימולטור הינו בסיס לבנית רשות מסוג זה.

### 7.3. מקורות:

<https://github.com/Microsoft/AirSim.git> •

[https://store.steampowered.com/app/365960/rFactor\\_2](https://store.steampowered.com/app/365960/rFactor_2) •