



پروژه پایانی دوره ی پردازش تصویر و مقدمه ای بر یادگیری ماشین

بهار ۲۰۲۱

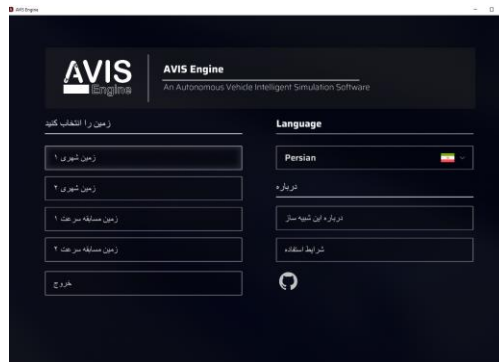
## فهرست

آشنایی با Avisengine .....	۳
نحوه ی پیاده سازی پروژه .....	۵
بخش امتیازی .....	۵
۱- موانع .....	۵
۲- تابلو ها .....	۵
۳- دیتاست .....	۵
۴- شبکه ی عصبی .....	۷
نامگذاری فایل ها .....	۸
نام فایل های کد .....	۸
فایل های بخش امتیازی .....	۸

## آشنایی با Avisengine :

در این پروژه قصد داریم با استفاده از چیز هایی که تا الان یاد گرفتیم ماشین ابتدا شبیه ساز را از سایت زیر دانلود کنید

<https://avisengine.com/>



شکل 1 فضای شبیه ساز

یک نقشه انتخاب کرده و موانع را میتوانید غیر فعال کنید (تشخیص موانع بخش امتیازی است)



شکل ۳ نقشه ی مسابقه ی سرعت



شکل ۲ نقشه ی زمین شهری

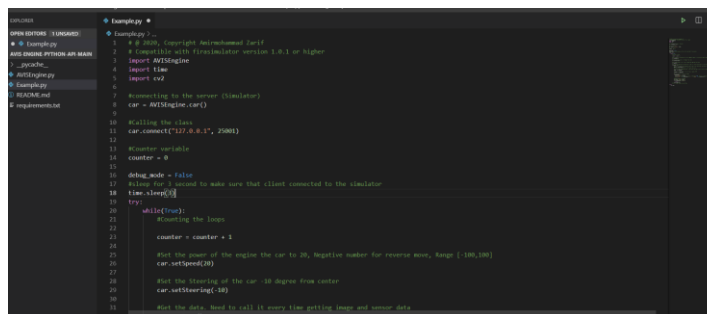
سپس میتوانید تنظیمات ماشین را بررسی کنید مانند سرعت و ..

برای اتصال کد و شبیه ساز باید در شبیه ساز شروع سرور را بزنید و سپس کد خود را ران کنید



شکل ۴ شروع سرور

پس از آشنایی با فضای شبیه ساز میتوانید فایل example.py را همراه با شبیه ساز ران کنید تا با نحوه ی کار کردن کد آشنا شوید.



شکل ۵ فایل مثال

همانطور که میبینید ابتدا آبجکتی از ماشین ساخته و سپس به سرور متصل میشویم

```
car = AVISEngine.car()
#Calling the class
car.connect("127.0.0.1", 25001)
```

سپس همانند کار با وب کم در یک while بینهایت تصاویر و اطلاعات ماشین را دریافت میکنیم

```
image = car.getImage()
# سنسور های ماشین
sensors = car.getSensors()
# تصویر دریافتی
image = car.getImage()
# سرعت ماشین
carSpeed = car.getSpeed()
```

و میتوان مثل قبل با استفاده از تصویر اطلاعات لازم را استخراج کرده و کارهای مناسب را بر اساس آن انجام دهید.

```
cv2.imshow('frames', image)
```

## نحوه ی پیاده سازی پروژه :

نقشه ای که به نظرتون آسوتتره انتخاب کنید.

۱- با استفاده از تمام دانسته های خود از کلاس ( مانند آستانه گیری، از بین بردن نویز) ابتدا فقط خطوط جاده رو مشخص کنید (برای مثال میتواند با استفاده از تشخیص رنگ زرد که در تمرین ۴ پیاده سازی کردید در نقشه مسابقه سرعت خط زرد را پیدا کرده و بر اساس آن برای حرکت ماشین تصمیم گیری کنید).

۲- حال با استفاده از Hough transformation مختصات خطوط را به دست آورده.

۳- سپس در صورت امکان ماشین را کنترل کنید.

## بخش امتیازی:

۱- موانع :

ماشین در برابر موانع نیز مسیر درست را انتخاب کند و به حرکت ادامه دهد.

۲- تابلو ها :

را تشخیص دهد و حرکت درست را انتخاب کند.

۱- میتوانید از qr کد های تصویر انتخاب کنید که برای این شبیه ساز کافی است.

۲- و میتوانید برای جامع تر کردن کدتون از از مدل های زیر که برای تشخیص علائم راهنمایی رانندگی هستن استفاده کنید.

<https://github.com/hoanglehaithanh/Traffic-Sign-Detection>

<https://github.com/Project-Road-Sign-Detection/Tensorflow-Street-Sign-Recognition>

۳- دیتاست:

همانطور که میدانید برای آموزش هوش مصنوعی نیاز به تعداد زیادی داده ی آموزش داریم.

در این موضوع که هوش مصنوعی قصد هدایت ماشین را دارد، داده های ما

۱- تصویر

۲- جهت فرمان ( برای مثال راست ۱ و بدون جهت ۰ و چپ - ۱ )

۳- گاز یا ترمز

است.

تعدادی از دیتاست هایی که محققان تهیه کردند در لینک زیر قابل دانلود است.

<https://analyticsindiamag.com/top-10-popular-datasets-for-autonomous-driving-projects/>

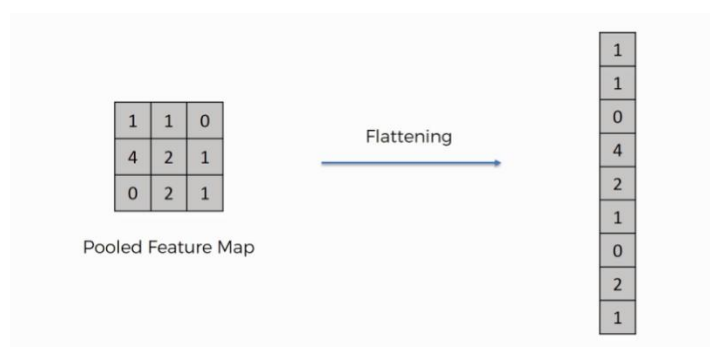
ما می‌خواهیم در این پروژه دیتاست خودمون رو ساخته و هوش مصنوعی با این دیتا آموزش دهیم.

برای اینکار باید مقداری در AVIS engine ماشین رو کنترل کنیم و فرمان ها و تصاویر را به صورت زیر ذخیره کنیم.

برای راحتی میتوانید فقط جهت فرمان را ذخیره کنید. و همینطور لازم نیست تمام فریم ها در حین بازی ذخیره کرد (چون حجم زیاد میگیره و همینطور خوب نیست و ممکن است باعث اورفیت شدن شود) میشه بعد از هر بیست فریم یک بار فرمان و فریم را ذخیره کرد که هم تنوع بیشتری از فریم ها و تصاویر داشته باشیم و هم حجم زیاد نشود.

نحوه ی ذخیره دیتاست:

هر تصویر را که یک ماتریکس  $(m,n)$  است تبدیل به یک بردار  $(m*n,1)$  کنیم



شکل ۶ تبدیل ماتریس به وکتور

سپس آن را با جهت مناسب فرمان در انتها در یک ردیف از فایل اکسل ذخیره کنیم برای مثال نمونه ای دیتاست در شکل زیر دیده میشود که پیکسل ها ذخیره و در انتهای آن لیبل تصویر ( یا جهتی که فرمان در آن تصویر باید حرکت کند ذخیره شده است)

pixels	Table
254 254 254 254 249 255 160 2 58 53 70 77 76 75 78 68 18 32 29 0 54 73 75 72 68 75 77 76 76 75 80 51 36 47 40 44 42 37 48 40 64 54 54 86 16 0 161 254 254 254 254 254 248 255 120 2 38 50 47 76 76 8	0
156 184 198 202 204 207 210 212 213 214 215 214 214 213 216 217 218 217 216 214 213 214 215 211 207 205 204 202 198 195 193 190 184 180 176 174 172 168 164 168 167 163 161 157 154 150	1
69 118 61 60 96 121 103 87 103 88 70 90 115 122 123 124 129 132 133 131 131 121 113 110 101 100 99 114 113 105 106 107 120 123 124 130 138 135 136 147 143 137 129 126 125 118 124 144 66 115 57	1
205 203 236 157 83 158 120 116 94 86 155 180 205 231 219 217 190 198 208 174 159 167 211 230 215 209 195 210 202 186 187 182 185 221 229 227 218 200 192 156 151 121 160 197 189 127 158 144 186	1
87 79 74 66 74 96 77 80 80 84 83 89 102 91 84 102 108 107 102 89 96 128 152 176 195 207 214 220 222 224 222 220 216 214 205 197 179 147 108 58 96 172 97 52 73 92 94 112 98 73 68 62 79 81 74 75 79 7	-1
235 223 223 109 34 37 34 31 28 38 56 69 106 136 153 163 145 135 136 127 152 158 152 144 138 121 65 38 56 42 34 31 28 33 35 39 40 34 29 32 39 50 39 25 16 24 28 37 231 233 220 75 22 44 31 39 36 44 70	0
71 70 104 147 166 170 195 145 156 154 146 129 139 130 117 103 104 107 111 101 90 79 75 110 126 101 79 89 95 113 107 111 105 102 129 126 148 156 171 196 175 148 159 153 154 108 118 116 45 80 83	1
176 177 170 168 173 171 167 169 166 139 98 107 121 136 138 141 154 155 160 155 153 143 135 121 101 103 101 76 27 13 17 17 17 22 45 59 75 76 59 55 38 67 74 83 92 94 87 108 181 180 180 176 172 174	-1
255 255	1
126 126 123 119 116 113 112 111 110 111 93 72 107 109 127 166 190 203 206 209 209 210 211 210 203 199 194 183 173 160 142 121 83 71 56 43 36 41 42 58 64 53 54 59 60 64 61 52 130 128 125 122 120	0
180 175 169 161 157 158 157 154 155 157 162 169 168 165 159 153 150 149 150 151 153 155 163 166 169 170 170 176 179 180 183 188 189 190 192 195 195 198 200 202 206 208 208 209 217 222 224 228	0
88 46 35 27 22 32 59 59 62 76 136 148 153 126 109 108 92 90 103 118 117 121 130 115 90 88 87 87 85 103 128 110 88 72 54 54 50 34 29 6 101 205 198 200 202 182 180 214 68 34 31 25 28 43 54 68 78 110	-1
121 112 64 104 101 87 118 74 91 128 89 109 91 27 127 197 191 186 189 192 194 197 195 192 190 186 178 177 169 161 151 133 105 84 52 29 24 21 22 23 20 16 12 14 18 25 89 80 143 109 76 125 93 111 114	1
165 203 211 204 216 204 202 194 195 191 207 209 196 202 209 214 214 215 193 186 175 128 180 208 160 130 189 215 188 169 184 201 184 190 203 173 193 202 203 203 205 207 204 195 176 163 174 175	1
22 28 27 28 26 28 31 33 33 30 32 23 19 44 75 110 120 127 138 138 151 155 129 128 127 125 118 117 119 96 109 94 65 54 18 13 22 26 26 27 26 19 22 21 20 18 20 18 25 29 27 24 26 32 31 37 34 27 23 41 93 1	0
132 154 165 176 182 187 192 199 203 206 208 212 216 221 223 223 222 220 218 217 214 209 207 210 212 206 202 197 192 192 194 195 192 186 177 173 163 151 142 92 70 89 139 141 63 139	1

شکل ۷ دیتاست

#### ۴- شبکه ی عصبی:

حال برای آموزش شبکه ی عصبی ابتدا فایل اکسل را خوانده و هر خط از این فایل اکسل شامل پیکسل ها و لیبل آن پیکسل ها ( جهت حرکت فرمان ) است این دو بخش را از هم جدا کرده و به ترتیب در متغیر X و Y بریزید برای مثال :

```
Y=df.iloc[:, -1]
X=df.iloc[:, :-1]
```

حال داده ها را به دو دسته تست و آموزش ( ۲۰ درصد داده ها برای تست در نظر گرفته شوند ) تقسیم کنید.

```
XTrain, XTest, YTrain, YTest = train_test_split(X, Y,
                                                test_size=0.2, random_state=10)
```

دقت کنید ما به سائز اولیه ی عکس ورودی به شبکه عصبی نیاز داریم.

حال شبکه ای برای آموزش درست کنید و بر اساس کد زیر برای آموزش این شبکه را آموزش دهید.

```
model = Sequential()

model.add(Convolution2D(24, (5, 5), (2, 2), input_shape=(XTrain.shape[0],
XTrain.shape[1], 1), activation='relu'))

model.add(Convolution2D(36, (5, 5), (2, 2), activation='relu'))
model.add(Convolution2D(48, (5, 5), (2, 2), activation='relu'))
model.add(Convolution2D(64, (3, 3), activation='relu'))
model.add(Convolution2D(64, (3, 3), activation='relu'))
```

```

model.add(Flatten())
model.add(Dense(100, activation = 'relu'))
model.add(Dense(50, activation = 'relu'))
model.add(Dense(10, activation = 'relu'))
model.add(Dense(3))

model.compile(Adam,loss='categorical_crossentropy', metrics=['accuracy
'])

history = model.fit(XTrain, YTrain,epochs=20,batch_size=64, validation
_data=(XTest,YTest))

#### STEP 9 - SAVE THE MODEL
model.save('model.h5')

```

حال بعد از سیو کردن مدل میتوانید با دادن هر فریم به این مدل برای جهت فرمان تصمیم گیری کنید.

## نامگذاری فایل ها :

نام فایل های کد:

- AutonomousVehicle.py

فایل های بخش امتیازی:

- Train.py
- Model.h5
- Dataset.csv

باشد

تمام فایل های حاوی

- کد
- گزارش

را به صورت یه فایل zip با نام گذاری به فرمت زیر دریاورید و ارسال کنید.

- اسمتون +Project
- Project\_MohmmadJavadRanjbar.zip
- Project\_MohammadJavadRanjbar.pdf