הגדלת הDATA :

* לקנות אוטובוסים דומים -
  + חרשתי את האינטרנט ולא מצאתי את הדגם הספציפי
* לחפש פתרונות באינטרנט לבעיות דומות - אין ממש פתרונות לבעיה כזו, בדרך כלל אומרים פשוט להשיג עוד דאטא מהנחה שאנחנו יצרנו את הDATA. כלל האצבע הוא 1000 דוגמאות למחלקה.
  + <https://towardsdatascience.com/image-augmentation-for-deep-learning-histogram-equalization-a71387f609b2>
  + <https://machinelearningmastery.com/image-augmentation-deep-learning-keras/>
  + חבילה מומלצת - keras
  + https://stackoverflow.com/questions/44906317/what-are-possible-values-for-data-augmentation-options-in-the-tensorflow-object
* לעשות פעולות קלאסיות על תמונות מהדאטה שקיים - סיבוב, סקאלה, הרעשה, תאורה, הסתרות.
* לבקש מהם ?

פתיחת תשתיות עבודה :

* GIT משותף - [https://github.com/matancor99/deep\_learning\_cv\_proj](https://github.com/matancor99/deep_learning_cv_proj" \t "_blank)
* One\_note

מחקר בהקשרי GOOGLE CLOWD

* https://github.com/tensorflow/models/blob/master/research/object\_detection/g3doc/running\_on\_cloud.md
* https://cloud.google.com/ml-engine/docs/getting-started-training-prediction

הבנת הארכיטקטורה של רשתות הנוירונים שנעשות בהן.

Cd C:\Users\matan\Desktop\cv\_proj\gcloud\cloudml-samples-master\census\estimator

הרצה לוקלית של ההרשת שלנו

gcloud ml-engine local train --module-name trainer.task --package-path trainer/ --job-dir output\ -- --train-files data/adult.data.csv --eval-files data/adult.test.csv --train-steps 1000 --eval-steps 100

הרצה לוקלית מבוזרת

gcloud ml-engine local train --module-name trainer.task --package-path trainer/ --job-dir output-dist --distributed -- --train-files data/adult.data.csv --eval-files data/adult.test.csv --train-steps 1000 --eval-steps 100

tensorboard --logdir=<output\_file\_name> --port=8080

**יצירת bucket :**

$PROJECT\_ID=$(gcloud config list project --format "value(core.project)")   
$BUCKET\_NAME="${PROJECT\_ID}-mlengine"

$REGION="us-central1"

gsutil mb -l $REGION gs://$BUCKET\_NAME

**Bucket data copying :**

gsutil cp -r data gs://$BUCKET\_NAME/data

$TRAIN\_DATA="gs://$BUCKET\_NAME/data/adult.data.csv"   
$EVAL\_DATA="gs://$BUCKET\_NAME/data/adult.test.csv"

gsutil cp ../test.json gs://$BUCKET\_NAME/data/test.json

$TEST\_JSON="gs://$BUCKET\_NAME/data/test.json"

$JOB\_NAME="census\_single\_1"

$OUTPUT\_PATH="gs://$BUCKET\_NAME/$JOB\_NAME"

RUN in the cloud command :

gcloud ml-engine jobs submit training $JOB\_NAME --job-dir $OUTPUT\_PATH --runtime-version 1.4 --module-name trainer.task --package-path trainer/ --region $REGION -- --train-files $TRAIN\_DATA --eval-files $EVAL\_DATA --train-steps 1000 --eval-steps 100 --verbosity DEBUG

REAL MODEL TRAINING :

1. Define the project and bucket name and paths.
2. Upload all the neccasery files to the data file in the bucket.
3. Define all the nessary param for the command.
4. Cd to tensorflow/models/research/
5. Run the command:

$PROJECT\_ID=$(gcloud config list project --format "value(core.project)")   
$BUCKET\_NAME="${PROJECT\_ID}-mlengine"

$REGION="us-central1"

$JOB\_NAME="buses\_training\_17"

$EVAL\_NAME="eval\_buses\_training\_17"

$TRAIN\_DIR="gs://$BUCKET\_NAME/$JOB\_NAME"

$EVAL\_DIR="gs://$BUCKET\_NAME/$EVAL\_NAME"

$PATH\_TO\_LOCAL\_YAML\_FILE = "C:/Users/matan/Desktop/cv\_proj/deep\_learning\_cv\_proj/training/cloud.yml"

$PIPELINE\_CONFIG\_PATH = "gs://$BUCKET\_NAME/data/faster\_rcnn\_inception\_v2\_pets.config"

gcloud ml-engine jobs submit training $JOB\_NAME --runtime-version 1.2 --job-dir $TRAIN\_DIR --packages "slim/dist/slim-0.1.tar.gz,dist/object\_detection-0.1.tar.gz" --python-version 2.7 --module-name object\_detection.train --region us-central1 --config $PATH\_TO\_LOCAL\_YAML\_FILE -- --train\_dir $TRAIN\_DIR --pipeline\_config\_path $PIPELINE\_CONFIG\_PATH

**Run from cloud shell – and after press wev preview on port 8080**

TRAIN\_DIR=gs://fair-catbird-199723-mlengine/buses\_training\_16

tensorboard --logdir=$TRAIN\_DIR –-port=8080

EVAL\_DIR=gs://fair-catbird-199723-mlengine/eval\_buses\_training\_16

tensorboard --logdir=$EVAL\_DIR –-port=8080

EVAL THE MODEL on the cloud:

gcloud ml-engine jobs submit training $EVAL\_NAME --runtime-version 1.2 --job-dir $TRAIN\_DIR --packages "slim/dist/slim-0.1.tar.gz,dist/object\_detection-0.1.tar.gz" --python-version 2.7 --module-name object\_detection.eval --region us-central1 --scale-tier BASIC\_GPU -- --checkpoint\_dir $TRAIN\_DIR --eval\_dir $EVAL\_DIR --pipeline\_config\_path $PIPELINE\_CONFIG\_PATH

EXPORT The MODEL (run from object detection):

python export\_inference\_graph.py --input\_type image\_tensor --pipeline\_config\_path "C:/Users/matan/Desktop/cv\_proj/deep\_learning\_cv\_proj/training/faster\_rcnn\_inception\_v2\_pets.config" --trained\_checkpoint\_prefix "C:/Users/matan/Desktop/cv\_proj/deep\_learning\_cv\_proj/data/model/model.ckpt-101096" --output\_directory inference\_graph

After this script runs we will have .pb file in the inference\_graph directory.

In Order to test the model we will run the Object\_detection\_image.py script – from the object detection folder.

Params to change in the script :

1. IMAGE\_NAME = 'test1.jpg' - line 34 – and move the image to the object detection file.
2. imS = cv2.resize(image, (960, 540)) - resize the image before display

**NOTE :**

**Things to do to be python 3.5 and tf 1.6 compatible again :**

1. Change the python version and runtime version in the command.
2. Change the runtime version in the cloud.yml file.
3. Remove lines 91-94 in the config file, delete the one in the cloud and re upload by - cd to the direcory of the config, and run: gsutil cp -r .\faster\_rcnn\_ince://$BUCKET\_NAME/data

1. Change the lines 168 of object\_detection\utils\learning\_schedules.py to:

**list(range(num\_boundaries)),**

**Remove the dist files in the research folder and in research/slim**

**Run python setup.py sdist in both folders and make sure we generate the dist files.**

1. **Follow the post about the unsolved issues.**