Eng. Ali Abbas  
Developer at LIT Company  
Tel: +963 (0)938 015 781  
ali.abbas@lit-co.net  
LIT LTD Akkar, Lebanon

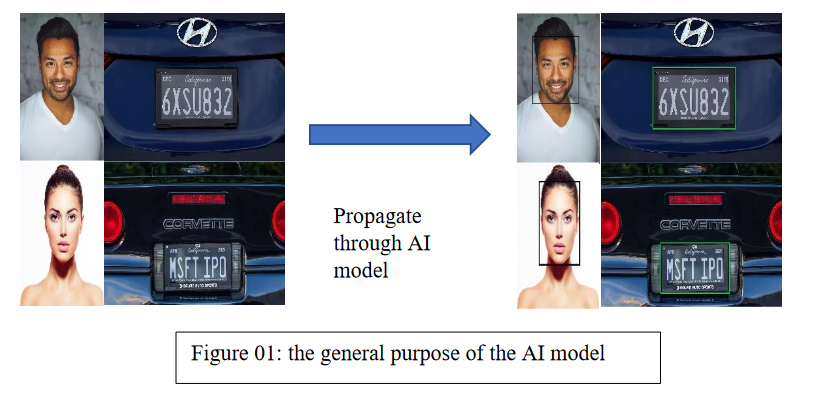
**AI Anonymization Project Report**

# **1) Overview:**

The General Data Protection Regulations (GDPR) enforces rules relating to the protection of natural persons with regards to the processing of personal data and rules relating to the free movement of personal data, to eventually protect fundamental rights and freedoms of natural persons and in particular their right to the protection of personal data.[1]

To comply with GDPR, companies have to redact information that expose personal details about individuals as stated in the GDPR manual, and these details include human faces and license plates.[2]

In that light, we think AI can do a great job in determining the position of such information in available data, so we planned a pipeline to pre-process the data using AI (i.e., automatically determining the location of face and/or license plates in an image.



The report discusses the project plan in the section 2, the plan details in article 2, subsection “a”, details of the progress and implementation in article 3.

# **2) Project Plan:**

In order to get a model that can accurately annotate images that might or might not contain faces and/or license plates, the process involves the following steps:

1. Dataset(s) selection
2. Data preparation
3. Model selection
4. Training
5. Evaluation
6. Candidate Model Selection

a) **Plan Details**:

**Data set selection**:

Face Detection Dataset: CelebA (205999 annotated faces). [3]

License Plates Detection Dataset: Chinses City Parking Data (350k annotated license plates). [4]

**Data preparation**: Datasets are saved on Google Drive, the preparation pipeline goes as follows: download -> extract -> read -> decode (from papers instructions) -> convert to COCO Format -> save info as JSON files to be instantly registered upon training.

**Model Selection**: we fine-tuned 6 pre-trained checkpoints from detectron2’s model zoo, that are

**Training**: Training parameters is fixed amongst all 6 trials.

**Evaluation**: Evaluate the trained models on Average Precision (AP) metric.

**Model Selection**: select the mode with the highest AP.

3) **Implementation Details**:

**Environment Setup**:

Python 3.8

detectron2 0.6

PyTorch 1.8 (and corresponding packages like torchvision and torchaudio)

numpy (1.24.2)

matplotlib (3.7.1)

gdown (4.7.1).

**Dataset Selection**: datasets are available on Google Drive (links are in the data-related references).

**Dataset Preparation**: the accepted architecture of a dataset to be input to a detectron2 model is as follows: List[dict(1st instance info), dict(2nd instance info), …, dict(ith instance info)], data is extracted after downloading to a directory named raw, and then each dataset information is written to a JSON file in COCO format to be used later in the data registration.

**Selected Pre-Trained Checkpoints**: we choose the following (models/variations):

faster\_rcnn\_R\_101\_FPN\_3x

faster\_rcnn\_R\_50\_DC5\_3x

faster\_rcnn\_R\_101\_C4\_3x

retinanet\_R\_50\_FPN\_3x

faster\_rcnn\_R\_50\_C4\_3x

retinanet\_R\_101\_FPN\_3x

**Training**: the parameters are fixed through all the training experiment:

**learning rate**: 0.00025 (decay at the 30kth step by a factor of 0.7)

**number of steps**: 60k

**batch size**: 1

**Evaluation**: summary of the training results showing the best performing model:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Monitor  Model | Localization Loss | Classification Loss | Total Loss | Average Precision (AP) |
| faster\_rcnn\_R\_101\_FPN\_3x | 0.0411 | 0.0168 | 0.0636 | 70.6 |
| faster\_rcnn\_R\_50\_DC5\_3x | 0.0353 | 0.0114 | 0.0566 | 72.11 |
| faster\_rcnn\_R\_101\_C4\_3x | 0.0373 | 0.0142 | 0.0744 | 71.76 |
| retinanet\_R\_50\_FPN\_3x | 0.153 | 0.0247 | 0.182 | 71.08 |
| faster\_rcnn\_R\_50\_C4\_3x | **0.0332** | **0.0119** | **0.0789** | **75.72** |
| retinanet\_R\_101\_FPN\_3x | 0.126 | 0.0198 | 0.156 | 68.98 |

Source: Training Statistics. [5]

Candidate Model Selection: **faster\_rcnn\_R\_50\_C4\_3x**

# **4) Final Step:**

Run training for the candidate model on a larger subset of the datasets (could be as large as 200k images per category and might take more than 5 days).

Execution of the final step can be done by running the following two commands

python src/prepare\_data.py

python src/train.py --yaml\_url COCO-Detection/faster\_rcnn\_R\_50\_C4\_3x.yaml

The reason why I didn’t train all potential checkpoints on the larger dataset is that The execution time of the training task is directly proportional to both the size of the dataset and the number of epochs, with an increase in either resulting in a corresponding increase in execution time. So, instead of waiting for more than a month to get the results I decided to go with a greedy approach and take the model that shows the most potential from the first epoch on a dataset that is relatively sufficient.

# **5) References:**

1. [Art. 1 GDPR – Subject-matter and objectives - General Data Protection Regulation (GDPR) (gdpr-info.eu)](https://gdpr-info.eu/art-1-gdpr/)
2. <https://github.com/understand-ai/anonymizer>
3. [Large-scale CelebFaces Attributes (CelebA) Dataset, Ziwei Liu, Ping Luo, Xiaogang Wang, Xiaoou Tang, Multimedia Laboratory, The Chinese University of Hong Kong (cuhk.edu.hk)](http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html)
4. [GitHub - detectRecog/CCPD: [ECCV 2018] CCPD: a diverse and well-annotated dataset for license plate detection and recognition](https://github.com/detectRecog/CCPD)
5. Training Statistics (Attached File).