DIABETIC RETINOPATHY SCORING USING GANS

BioInformatics - Project 9
Politecnico di Torino

Matteo Abbamonte
Alkis Koudounas

OUTLINE

- Diabetic Retinopathy Detection Dataset
- GAN architectures
- Features Extractor
- Linear Regressor
- Performance Evaluation

DIABETIC RETINOPATHY DETECTION DATASET

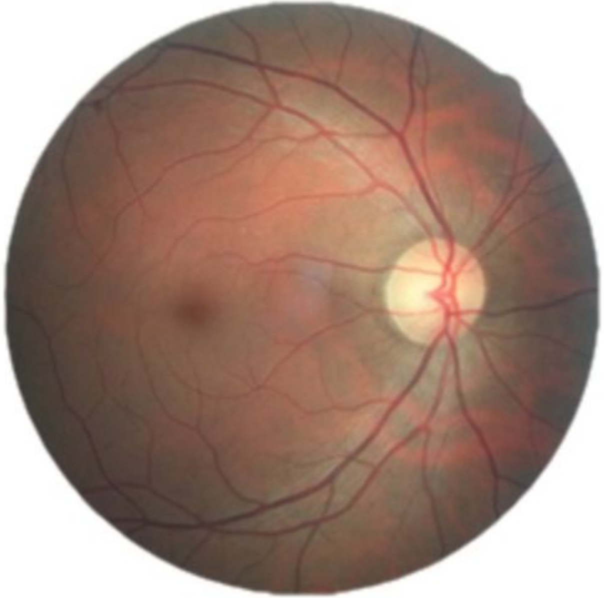
The Kaggle's Diabetic Retinopathy Detection dataset is a large set (83.23 GB) of high-resolution retina images taken under a variety of imaging conditions.

A left and right field is provided for every subject, and images are labeled with a subject ID, as well as either left or right. The images in the dataset come from different models and types of cameras, which can affect the visual appearance of left vs. right.

Like any real-world data set, there is noise in both the images and labels: images may contain artifacts, be out of focus, underexposed, or overexposed.

A clinician has rated the presence of diabetic retinopathy in each image on a scale of 0 to 4, according to the following scale:

- 0 : No DR
- 1: Mild
- 2: Moderate
- 3: Severe
- 4: Proliferative DR



BTGRAHAM-300

Benjamin Graham propose a solution to pre-process these images in order to remove some of the variation between the images due to differing lighting conditions, camera resolution, etc.

The steps he performs are:

- Rescaling the images to have the same radius, that is 300 pixels or 500 pixels (we used the version with 300 px)
- Subtracting the local average color; the local average gets mapped to 50% gray
- Cropping the images to 90% size to remove the "boundary effects"

Image: 13_left
Rating: 0

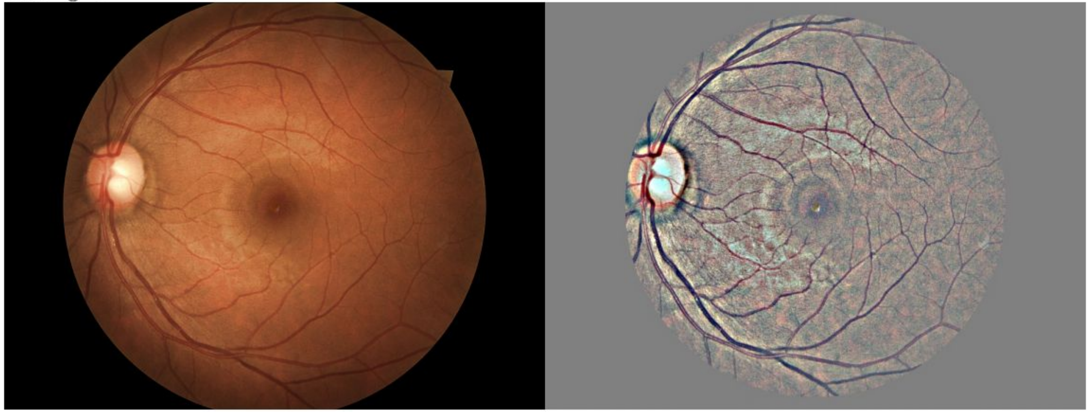
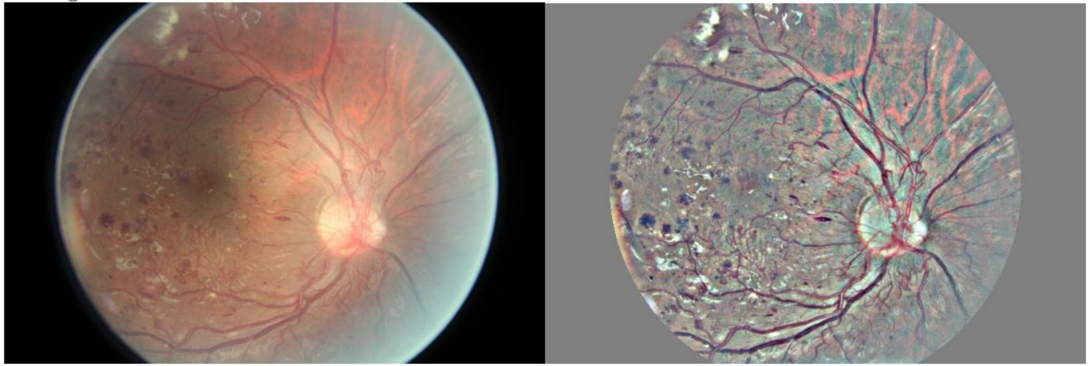


Image: 16_left
Rating: 4

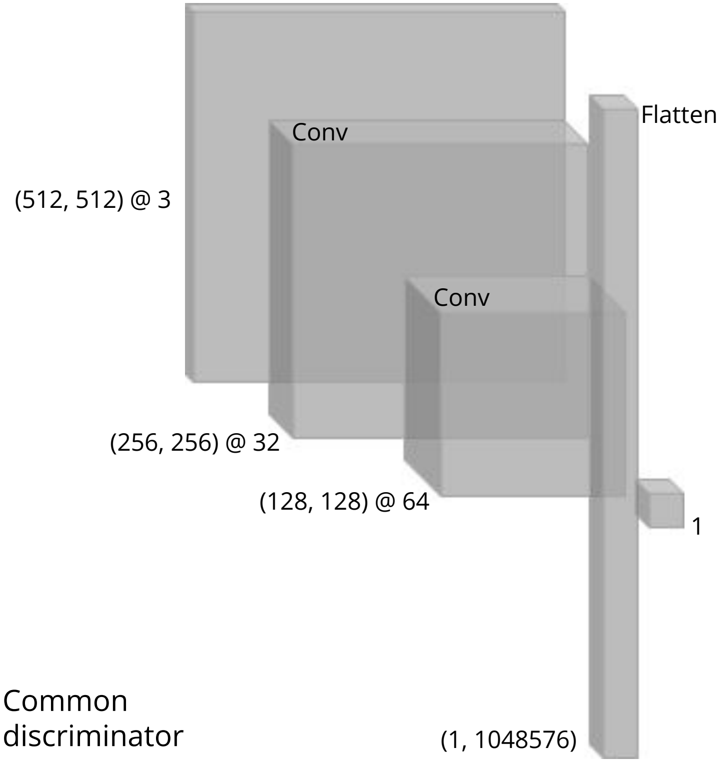


GAN ARCHITECTURES

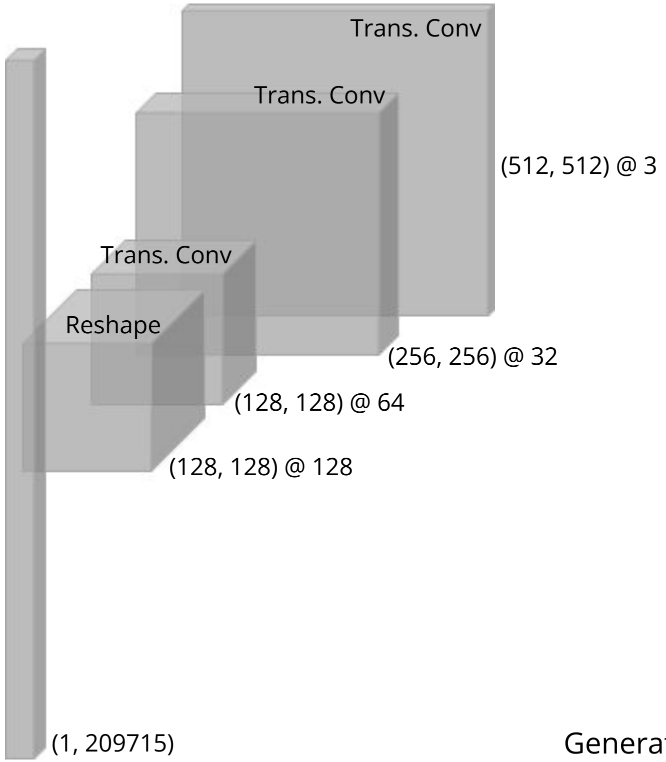
To achieve the goal of the project, two different DC-GAN architectures are proposed. For each GAN three steps are performed, with increasing network depth.

The discriminator of both the GANs presents in the first step the simplest structure while in the third one a more complex yet more effective one.

First step - first GAN

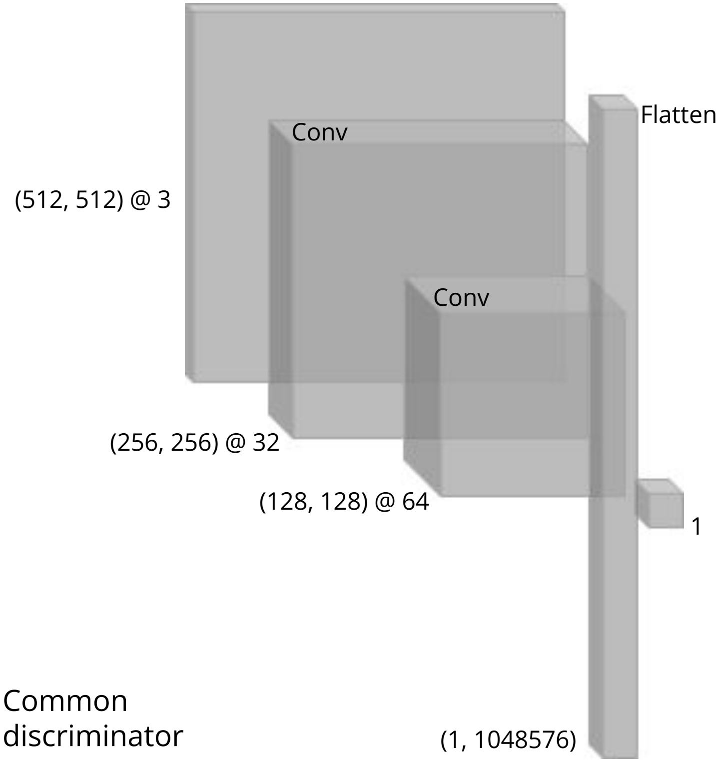


Common
discriminator

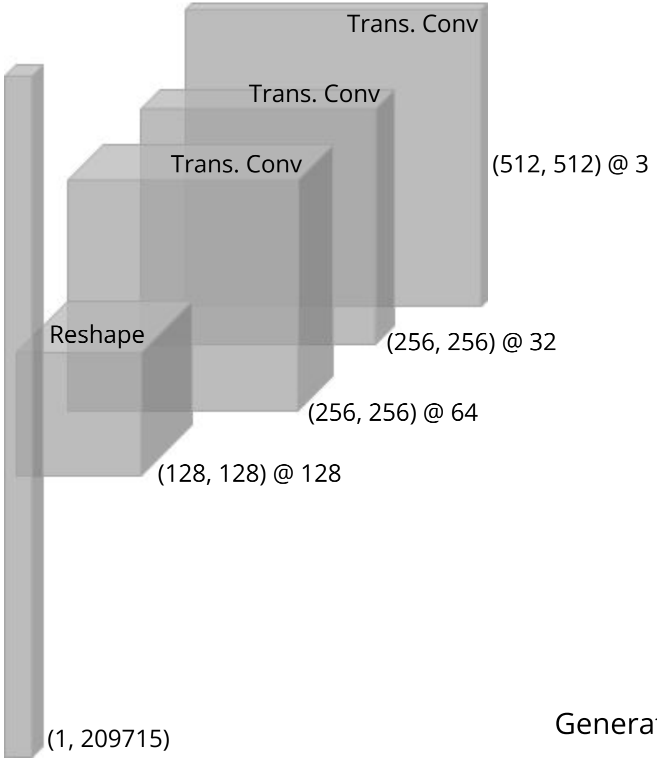


Generator

First step - second GAN

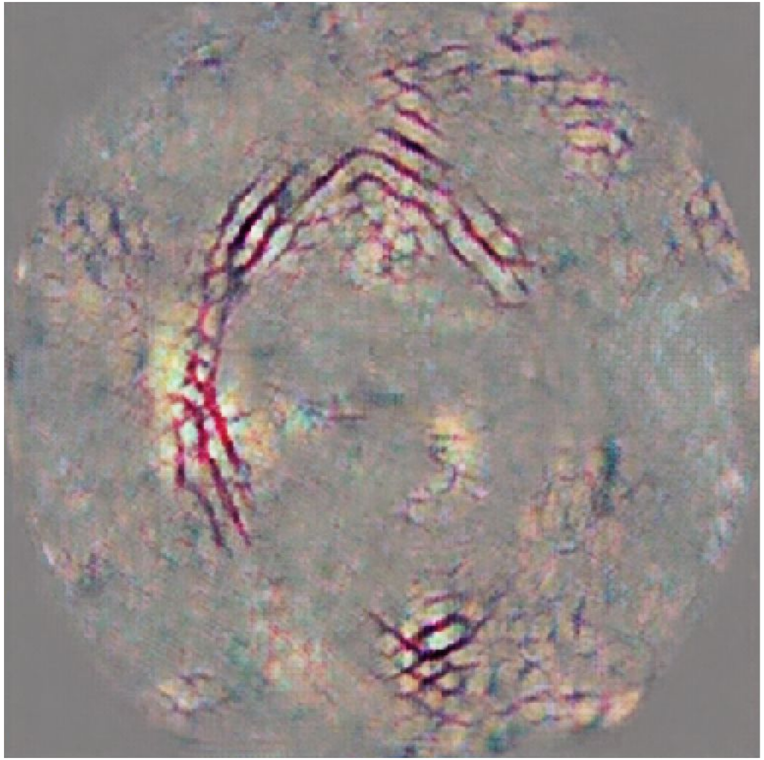


Common
discriminator

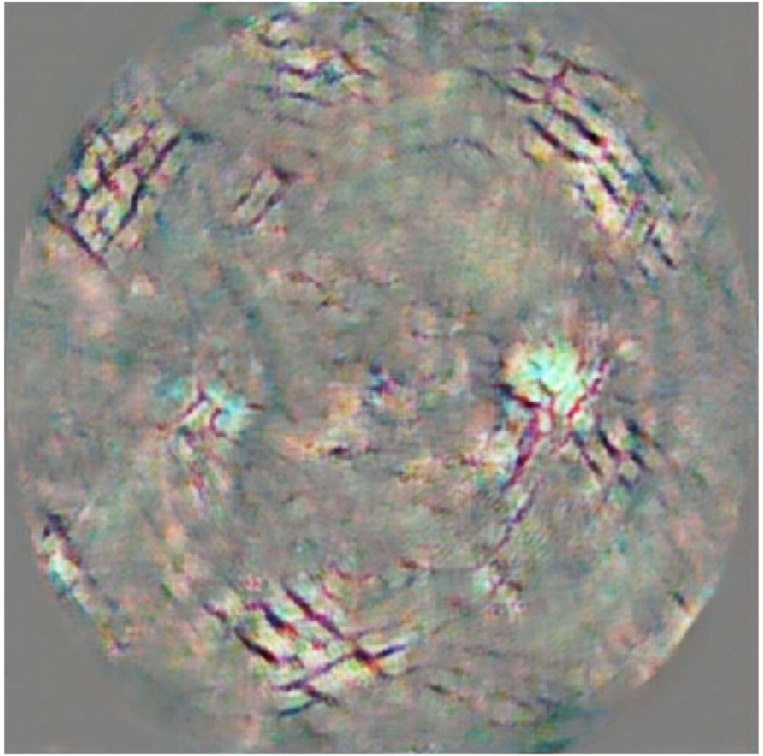


Generator

First step - Results

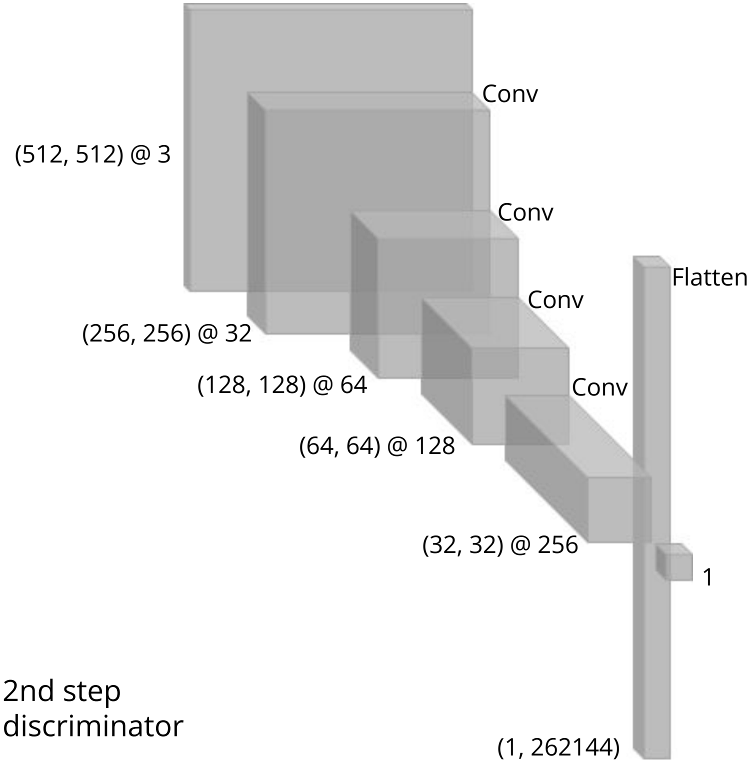


First GAN sample

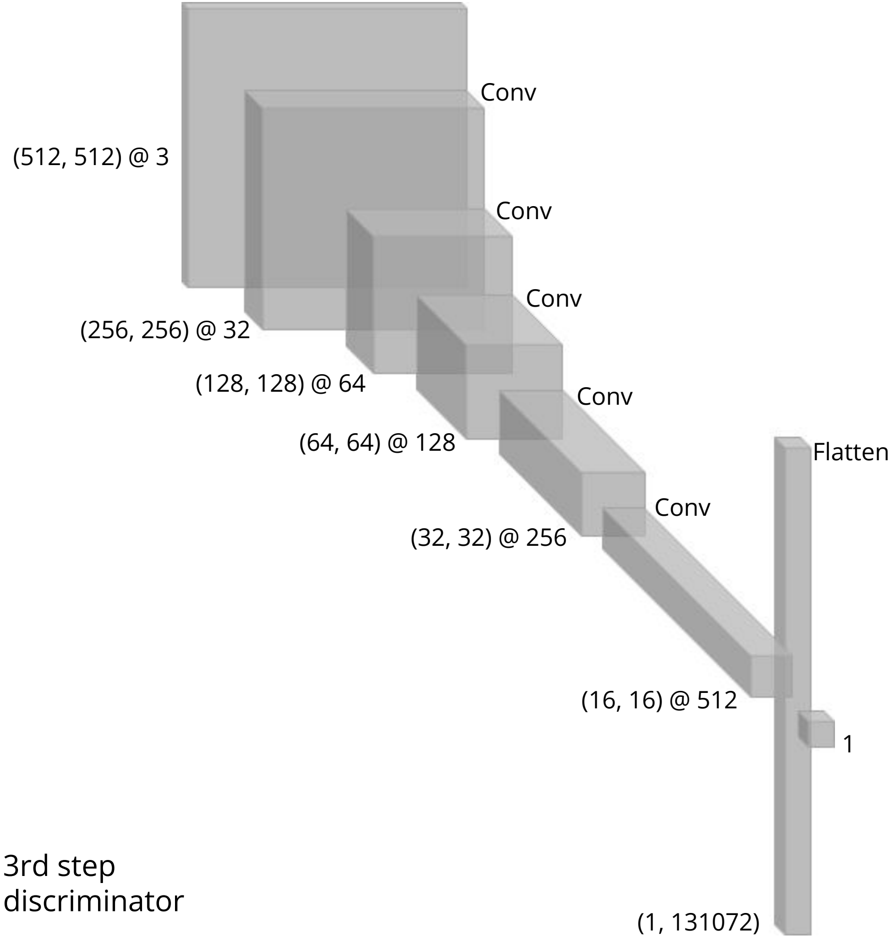


Second GAN sample

Further steps

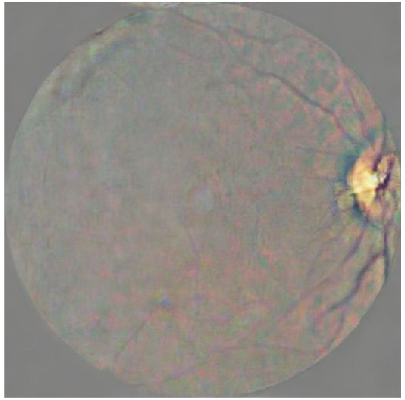


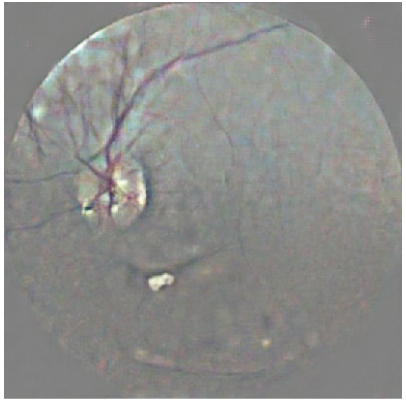
2nd step
discriminator



3rd step
discriminator

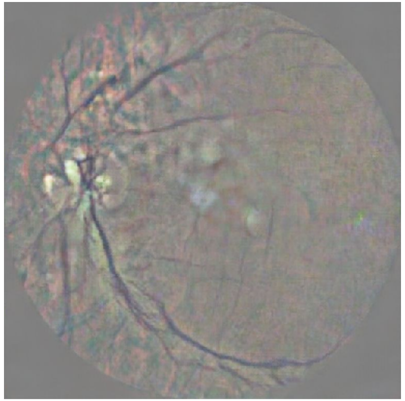
Further steps - Results

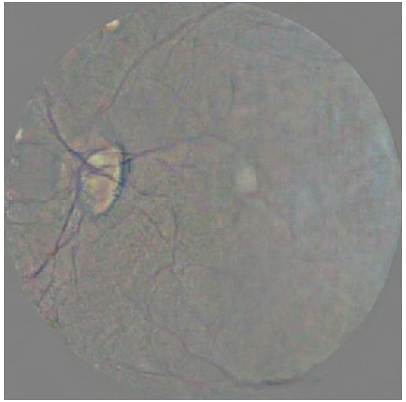




Samples from First and Second GAN - second step

Samples from First and Second GAN - third step





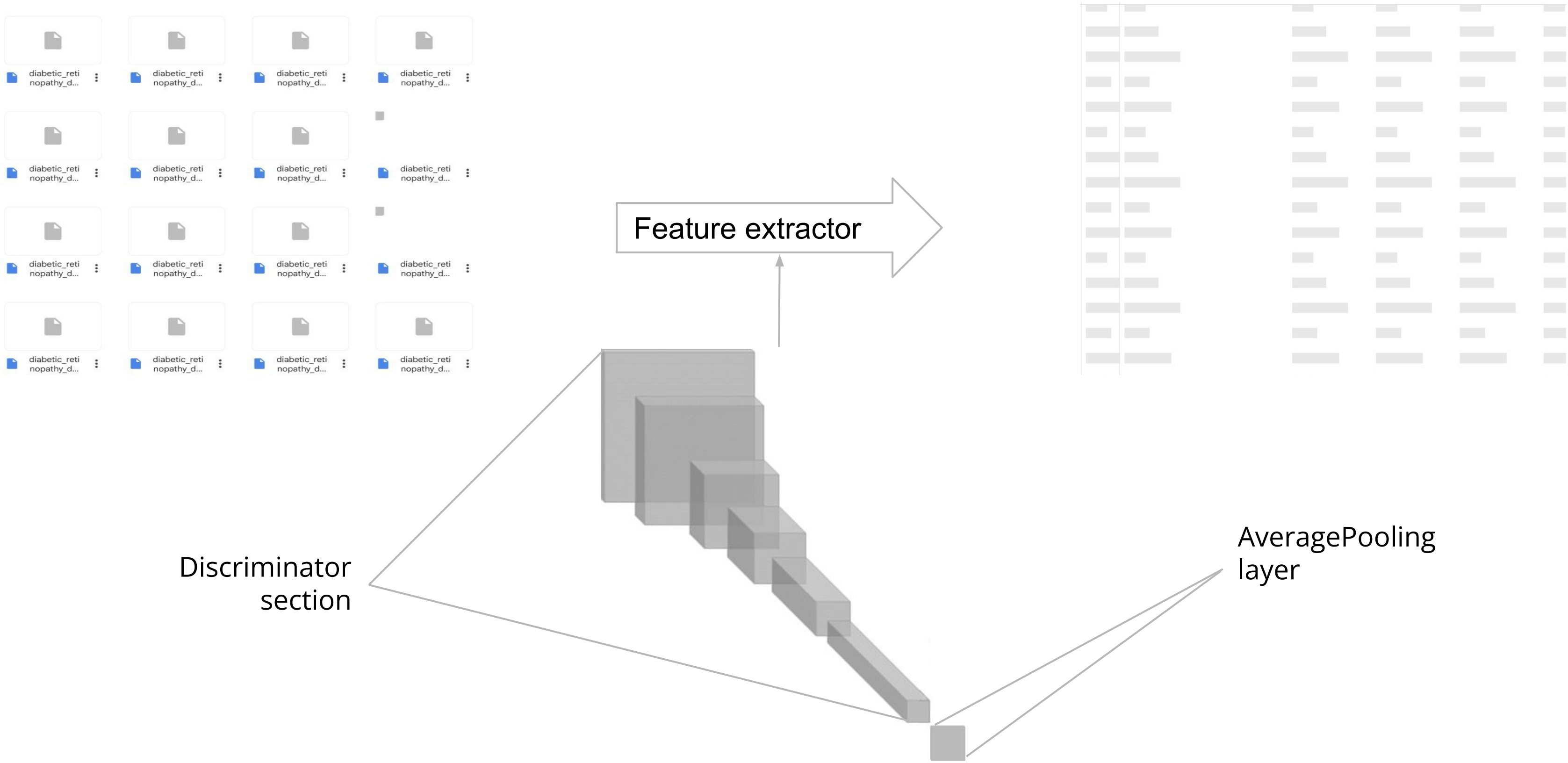
FEATURES EXTRACTOR

To extract the features by using the GAN Discriminator, we took the layers of the network until the fully-connected blocks, and we applied a global average pooling layer to further reduce the number of features to a more appropriate order of magnitude.

Thus, we applied this feature extractor to the original dataset to create the corresponding features dataset.

This process is followed for each of the previously defined GAN architectures, by considering the third step setup.

Features extractor

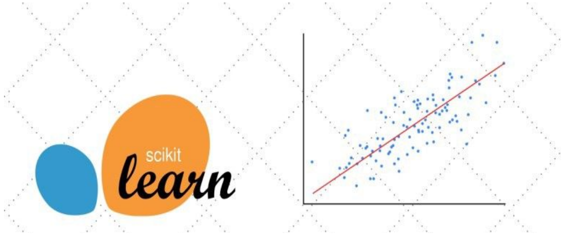


LINEAR REGRESSOR

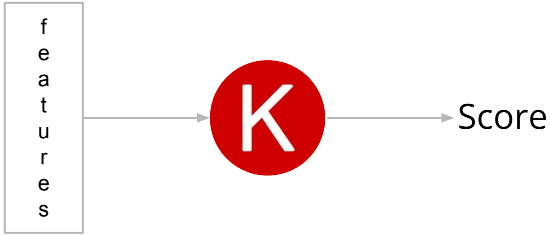
To predict the clinical score from the features just obtained, we built three different linear regression models, using SKLearn implementation, an analytical solution and a model consisting of a single Keras Dense layer with a single neuron (unit).

Since the dataset structure is inherently unbalanced (due to the preponderance of samples labelled with “class” -score- 0), we decided to apply a class weighting strategy.

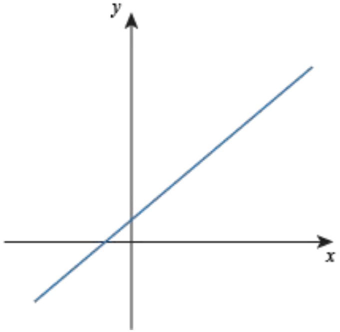
Linear regression models



SciKit Learn Linear
Regression model



Keras Dense Layer



Analytical parameters
optimization

score = M X + q

where:

- X: input features
- M: angular coefficient array
- q: constant term

PERFORMANCE EVALUATION

Performance evaluation - Cross Validation

Linear Regressor Models		Stratified K-Fold Cross Validation	
		Validation Loss	Computation Time (s)
1st GAN	**SKLearn**	**0.9889**	**1.0928**
	Analytical	1.0508	83.8348
	Dense Layer	1.0691	245.2631
2nd GAN	**SKLearn**	**0.9778**	**1.0769**
	Analytical	1.0037	74.8295
	Dense Layer	1.0212	271.0011

K = 5
Training epochs = 2000

Performance evaluation - Metrics

Linear Regressor Models		Performance Evaluation	
		MAE	RMSE
1st GAN	**SKLearn**	**1.1217**	**1.3095**
	Analytical	1.1290	1.3155
2nd GAN	**SKLearn**	**1.1157**	**1.3005**
	Analytical	1.1172	1.3011

Training epochs = 10000

Performance evaluation - Predictions

```
GT Label: 0      SKLearn score: 0.935626      Analytical Optimization score: 0.945135
GT Label: 0      SKLearn score: 0.503648      Analytical Optimization score: 0.441096
GT Label: 1      SKLearn score: 1.030144      Analytical Optimization score: 1.115257
GT Label: 2      SKLearn score: 0.819170      Analytical Optimization score: 0.858270
GT Label: 0      SKLearn score: 0.885946      Analytical Optimization score: 0.877235
GT Label: 0      SKLearn score: 2.605192      Analytical Optimization score: 2.582366
GT Label: 1      SKLearn score: 1.716710      Analytical Optimization score: 1.719398
GT Label: 0      SKLearn score: 0.502255      Analytical Optimization score: 0.536965
GT Label: 0      SKLearn score: 1.580807      Analytical Optimization score: 1.581999
GT Label: 0      SKLearn score: 0.608697      Analytical Optimization score: 0.572440
GT Label: 0      SKLearn score: 1.885379      Analytical Optimization score: 1.875245
GT Label: 2      SKLearn score: 1.213920      Analytical Optimization score: 1.210414
GT Label: 0      SKLearn score: 2.682424      Analytical Optimization score: 2.668019
GT Label: 0      SKLearn score: 0.902940      Analytical Optimization score: 0.927613
GT Label: 0      SKLearn score: 1.882490      Analytical Optimization score: 1.914978
GT Label: 0      SKLearn score: 1.684250      Analytical Optimization score: 1.685880
GT Label: 0      SKLearn score: 0.204668      Analytical Optimization score: 0.124225
GT Label: 0      SKLearn score: 1.563274      Analytical Optimization score: 1.574678
GT Label: 0      SKLearn score: 0.691172      Analytical Optimization score: 0.732723
GT Label: 0      SKLearn score: 1.692190      Analytical Optimization score: 1.665547
GT Label: 2      SKLearn score: 0.495732      Analytical Optimization score: 0.507677
GT Label: 0      SKLearn score: 0.891651      Analytical Optimization score: 0.856122
GT Label: 0      SKLearn score: 0.333551      Analytical Optimization score: 0.302627
GT Label: 0      SKLearn score: 1.249180      Analytical Optimization score: 1.269407
GT Label: 0      SKLearn score: 2.413505      Analytical Optimization score: 2.433668
GT Label: 0      SKLearn score: 0.802732      Analytical Optimization score: 0.822704
GT Label: 0      SKLearn score: 1.025987      Analytical Optimization score: 1.016611
GT Label: 3      SKLearn score: 0.645294      Analytical Optimization score: 0.693031
GT Label: 0      SKLearn score: 1.687023      Analytical Optimization score: 1.640098
GT Label: 2      SKLearn score: 2.636037      Analytical Optimization score: 2.682354
```

GAN 2 based feature extraction

THANK YOU