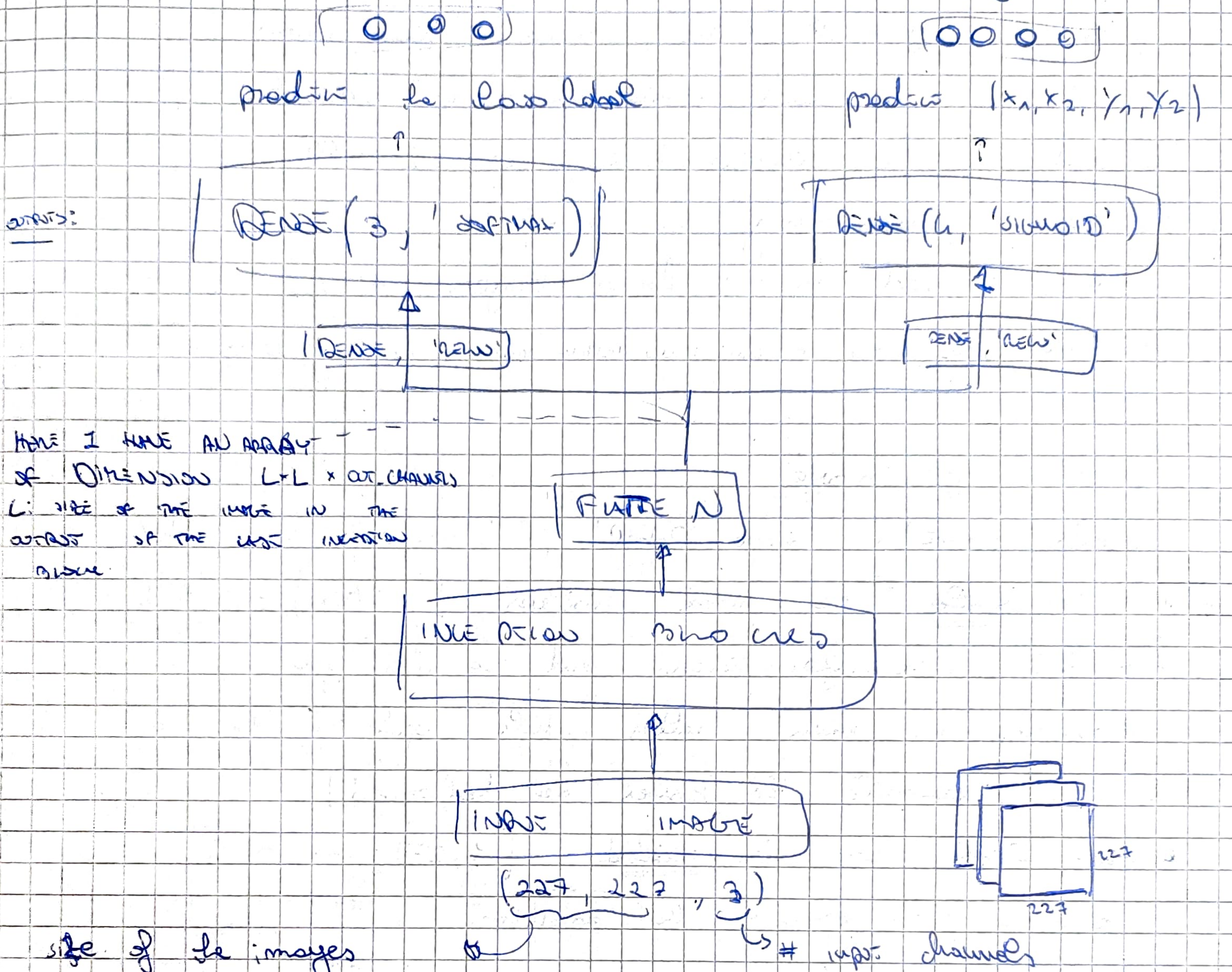
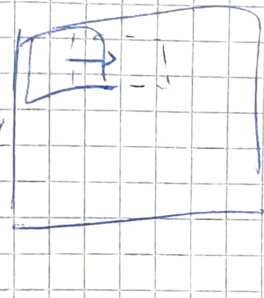
ALESSANDRO GHIOTTO S13944



1. CONVOLUTIONAL NEURAL NETWORK:

In such a task we need to detect a pattern within the whole input (MLP doesn't work, since it matches the whole input). Instead CNN are shift invariant, we have a little model which analyze each window of the image.

CONVOLUTIONAL LAYERS: works like a sliding filter that analyze each window of the image, and as output gives the features extracted



POOLING: select the features and decrease the dimension of the input.

INCEPTION BLOCK: have a multi-scale view of the input since for the same input, we use different filter size (which later are combined together)

big filter → general view, more "far away"

small " → I look more at the details of the input

At the end of the convolutional part I will have the features "describing" the image.

↳ that we will feed by 2 dense layers to get the outputs

[IT IS PARTICULARLY USEFUL WHEN WE DON'T KNOW NOTHING ABOUT THE SIZE OF THE OBJECTS IN THE IMAGES]

2\. a) Input: "input_image"

Labels (target): "image_label", "bounding_box_label"

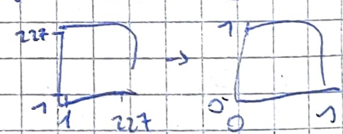
b) • Min Max scale the pixel values $[0, 255] \rightarrow [0, 1]$

↳ we don't lose information and we don't give too much importance to high values while updating the weights

(we don't oversample, the classes are balanced)

• bbox labels are also scaled $[1, 227] \rightarrow [0, 1]$

value in $[0, 1]$ represent the "proportion" of the corner (0 → ORIGIN, 1 → MAX DISTANCE)



↳ this allows me to use a sigmoid as output

• {1,2,3} labels are kept as they are

c) each sample have dimension (227, 227, 3), each pixel have value in $[0, 1]$

(then at train we will have another dimension indicating the batch size)

3. • 'label' → {1, 2, 3} , so as output I use a dense layer with 'softmax' activation function and 3 neurons
the softmax create a prob. distribution over the 3 classes which is the ideal for multiclass classification.
then to pick the actual label we can just use the 'argmax'
• b. boxes → (x_1, x_2, y_1, y_2), with each entry belonging to [0, 1]
so I can use a 'sigmoid' (with 4 neurons)
↳ one for each value (x_1, x_2, y_1, y_2)
↳ (I have 2 outputs)

4. for 2 outputs we need 2 losses.
• 'sparse_categorical_crossentropy' for the labels
categorical crossentropy: minimize the Kullback-Leibler divergence. so it makes the distribution of the output (which was a softmax) closer and closer to the distribution of the true labels.
sparse → just like the "normal" one, but works with indexes {1, 2, 3} instead of one-hot vectors $\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$

• 'b. boxes' → 'mse'
↳ just a regression in [0, 1]
we are minimizing the ~~dist~~ squared distance between the predicted value and the true coordinates

5. a) INPUT → INCEPTION BLOCKS → FLATTEN → DENSE (~~[illegible]~~ RELU) → DENSE (SOFTMAX)

→ DENSE (~~[illegible]~~ RELU) → DENSE (SIGMOID)

b) as activation I use the 'ReLu', since I don't need a bounded function, Relu is simple and works well with CNN (I should not have much problems of exploding gradient like in RNN)

softmax and sigmoid as output

c) • Initializer → He (for the convolutional blocks)

[set a good initial condition across the conv. layer, with the Relu activation function]

• As a form of "regularization" I use dropout layers that by randomly turning off some neurons, help the NN to don't ~~[illegible]~~ take decisions based solely on single neurons.

d) 1) number of inception blocks

the number required could depend on the complexity of the images

2) I try at least some value of batch size, for getting a good training of the model

3) I increase the dropout rate if I see overfitting

* [I ADD TWO DENSE LAYERS FOR DECREASING THE DIMENSION OF THE FLATTENED VECTOR BEFORE OF THE OUTPUT (LIKE A SMALL MLP)]

6) I would split the train and test (something like 80:20)

since we don't have many sample, we can't take a test set too little.

then apply cross validation on the train set.

C.V. is particularly useful (w.r.t. ~~[illegible]~~, hold out) when we don't have a lot of data.

metrics → accuracy

(I don't have a real need of using a F-score, since the classes are balanced)