Sapienza Università di Roma

Advanced Machine Learning

Final project

Image classification with Shannon information.

Students

Mert Yildiz
1951070
Ali Reza Seifi Mojaddar
1900547
Gianmarco Ursini
1635956
Mohammadreza Mowlai
1917906

Professor

Prof. Fabio Galasso

Accademic Year 2021/2022

Introduction

Recognition of images is a simple task for humans as it is easy to distinguish between different features. Somehow human brains are trained unconsciously with different or similar types of images that have helped human to distinguish between features (images) easily. However, machines need a lot of training for feature extraction which becomes a challenge due to high computation cost, memory requirement, and processing power. Nowadays, many new applications of image classification has been tried to decrease the memory requirement, computation cost and increase the accuracy of the classification of images.

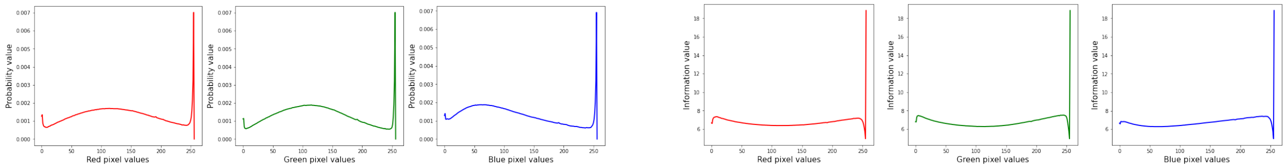
In this project a new approach on image classification has been applied from scratch. The main purpose is to try to increase the classification accuracy with a method that has never been applied, i.e. leveraging the usage of pixel's values Shannon informations (SI) in two different ways. Besides applying these approaches with a simple Convolutional Neural Network (CNN), in case of over fitting regular dropout and Monte Carlo(MC) dropout has been applied. In the end, a comparison between all the implemented models results has been performed.

The Dataset

In this project, CIFAR-100 dataset (Canadian Institute for Advanced Research, 100 classes) has been used. CIFAR-100 dataset is a subset of the Tiny Images dataset and consists of 60000 32x32x3 color images. The 100 classes in the CIFAR-100 are grouped into 20 superclasses. There are 600 images per class. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs) which the project classified images w.r.t. the superclass. There are 500 training images and 100 testing images per class, in total 50,000 training and 10,000 test images. The training set has been split into train and validation. Therefore, the classifier has been trained only on training set, hyper-parameters selected based on validation set and the unseen test set has been used to evaluate the model.

Shannon information

The Shannon information for an event E is defined as $\text{S.I.}(E) = \log\left(\frac{1}{\mathbb{P}(E)}\right)$. It is infact clear that if an event E has a really low probability to happen (e.g. the sentence "it's august and it's **snowing** in Palermo"), the event E itself will be may more informative w.r.t. an event E' with an high probability to happen (e.g. the sentence "it's august and it's **hot** in Palermo" is so common that the amount of information added by E' to our knowledge is negligible). Equally, it is possible to evaluate the Shannon information expressed by red, green and blue pixel values (denoted by V_{red}, V_{green} and V_{blue}) computing the probability, for example, for V_{red} to be in a certain interval (running this computation only across the training dataset). Being V_{red}, V_{green} and V_{blue} floating point values, we discretized the pixel values support into bins of size 1 (i.e. we will compute $\mathbb{P}(0 \leq V_{\text{red}} \leq 1)$, $\mathbb{P}(1 \leq V_{\text{red}} \leq 2)$ and so on). Aim of the project is to assess if the usage of the so computed Shannon information can somehow trigger a certain kind of attention of our model towards the most rare/informative pixels in order to improve its accuracy [1].



(a) Probability distributions for red, green and blue pixel values.

(b) S.I.(V_{red}), S.I.(V_{red}), S.I.(V_{red}) for the different bins.

[1] Even if red and green pixel values probability distributions looks similar, the blue one looks skewed and can be exploited to extract the desired accuracy improvement.

Our model

Since we are just focusing on the relevance of the Shannon information on improving accuracy, a simple CNN will be used. It contains 5 convolutional blocks (each containing a Convolutional layer, a MaxPooling layer with kernel of size $(2, 2)$ and a ReLu activation layer) each of them containing respectively 128, 512, 512, 512 and 512 filters. A dense layer with 20 neurons is then appended at the end as classifier. As loss function, Categorical Crossentropy has been used.

1° approach: appending Shannon informations to images

In this first approach, each image is transformed from a tensor of size $(32, 32, 3)$ into a tensor of size $(32, 32, 6)$, i.e. for each pixel in the RGB layers the image will be augmented to store its Shannon information value in the corresponding Shannon information layer.

2° approach: a Shannon information regularization term

In this second approach, original images of size $(32, 32, 3)$ are fed into the CNN. The Shannon information content is instead exploited by the addition of a regularization therm R.T.$(\cdot)$ to the Categorical Crossentropy loss. Denoting with $\widehat{y^{(i)}}$ the CNN label prediction for the i_{th} image, with $y^{(i)}$ its ground truth label, with λ the variable modulating the regularization intensity and with S.I.$^{(i)}$ the mean of the Shannon informations for the pixels contained in the i_{th} image, our custom regularization therm is defined as follows:

$$\text{R.T.}(\text{S.I.}^{(i)}) = \lambda \cdot (y^{(i)} - \widehat{y^{(i)}})^2 \cdot \text{S.I.}^{(i)}$$

The square value of $(y^{(i)} - \widehat{y^{(i)}})$ has been used since we want the regularization therm to influence the update of the CNN weights during the backpropagation phase only if $y^{(i)} \neq \widehat{y^{(i)}}$. The idea behind is that we want to penalize our model in an heavier way if a misclassification occured on top of an image containing very informative pixels w.r.t. a misclassification occurred over an image containing very common pixel values. Since it can be argued that regularizations offer in general a way to improve the test accuracy, a comparison will be run between our Shannon regularized model and the benchmark L_2 regularized one.

Dropout and MCDropout

In this approach Dropout layers are added to the previous model in order to evaluate the overall performance over the train-set and test-set. Generally by using Dropout, each time the model's architecture is slightly different and the outcome will be as an averaging ensemble of many different neural networks, each trained on one batch of data only. The difference between the regular Dropout and MC-Dropout is that dropout will be on even during test time. Then, instead of one prediction, there will be many predictions. The results of this section are summarized in Table 1, but generally MC-Dropout tends to lead to more accurate predictions, which additionally express the model's uncertainty.

Results and comparison

As it has been explained and described in the previous sections there are different approaches that have been applied. The CNN model has been applied on normal images dataset without any dropout, with regular dropout and MC Dropout. After creating augmented dataset with SI the same has been applied on augmented dataset, normal images dataset with SI regularization term and lastly the model has been applied with normal images dataset with L2 regularization without any dropout, with regular dropout and MC Dropout.

Dataset	**Regularization**	**Dropout**	**Test set accuracy**
Normal Images	None	None	52,34 %
Normal Images	None	Regular	53,14 %
Normal Images	None	MC Dropout	59,86 %
Augmented Dataset	None	None	50.32 %
Augmented Dataset	None	Regular	49.88 %
Augmented Dataset	None	MC Dropout	56.09 %
Normal Dataset	Shannon Regularization ($\lambda = 0.01$)	None	54,58 %
Normal Dataset	Shannon Regularization ($\lambda = 0.01$)	Regular	54,03 %
Normal Dataset	Shannon Regularization ($\lambda = 0.01$)	MC Dropout	60,48 %
Normal Dataset	L_2 Regularization ($\lambda = 0.0005$)	None	55,96 %
Normal Dataset	L_2 Regularization ($\lambda = 0.0005$)	Regular	55,66 %
Normal Dataset	L_2 Regularization ($\lambda = 0.0005$)	MC Dropout	61,44 %

Table 1: Comparisons of the Models Test Accuracy.

.

After running all approaches the accuracy on test set were as shown on the table above. As we can see the model applied with normal images dataset is outperforming the model with augmented dataset for all models however the model applied over the normal dataset that uses Shannon regularization (SR) term is outperforming the model applied over the normal images dataset without any regularization. Surprisingly, the models with L2 regularization are outperforming all the others.
The train, validation accuracy and loss plots has been added in the appendices showing that the models were over-fitting without the dropout and it has been reduced with dropout. Also to reduce the over-fitting it can be seen that the regularization term can be used.

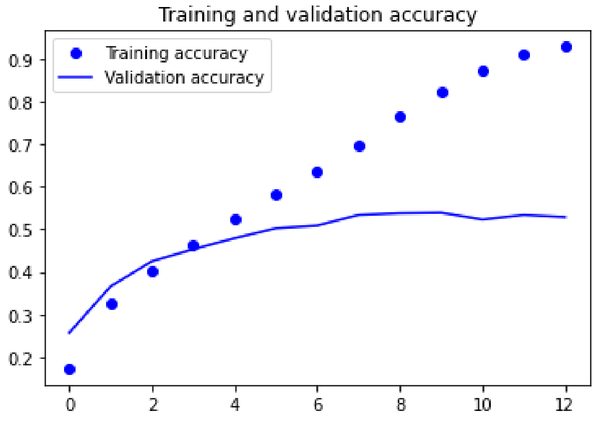
Conclusions

As we noticed, SI is not resulting into a better accuracy when appended to images as layers. Indeed, we concluded that 1° approach is just resulting into the production of a noisier dataset. In 2° approach we leveraged SI including it in our custom SI regularization term. At a first glance seems clear that the usage of the SR helps to improve the test set accuracy (i.e. to reduce the overfitting). Unluckily it resulted that an L_2 regularized model is performing better than the SR one. Indeed, we can claim that an improvement with the SR occurred because regularizations help in general to reduce overfitting, but there is no evidence that allows to consider the SR more effective than a common benchmark regularization (e.g. L_2 regularization). On the other hand, we definetely concluded that Dropout helps to reduce overfitting and that MC Dropout leads to more accurate predictions since it dumps model's predicting uncertainty.

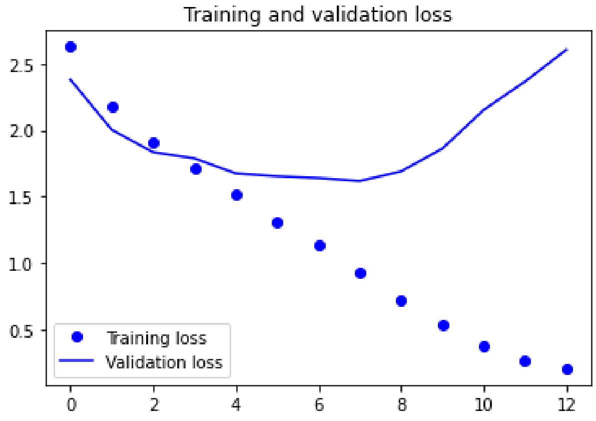
Future works

Our results leads us to claim that no meaningful usage of the SI has been found to increase the model accuracy. We want to recall the fact that our SI analysis might be biased by assuming that the events V_{red}, V_{green} and V_{blue} are mutually **independent** (we computed the probabilities to meet a certain pixel value for a color regardless of the other color's pixel values in the same spatial position). A first way to override this bias into a future work might be the application of both SI approaches on top of a dataset only made by BlackWhite images. In this way, since a single layer will be used, no combinations of pixel values has to be considered. A second way might be the evaluation of the logical union of the events V_{red}, V_{green} and V_{blue}. Of course, since using in this framework a `bin-size`=1 to compute probabilities would result into 255^3 possible combinations of pixel's values and we would meet the same V_{red}, V_{green} and V_{blue} combination on average every $\sim 20 \cdot 10^3$ pictures, a grosser `bin-size` value would be needed, and this might possibly result into a too coarse approximation of pixel values.

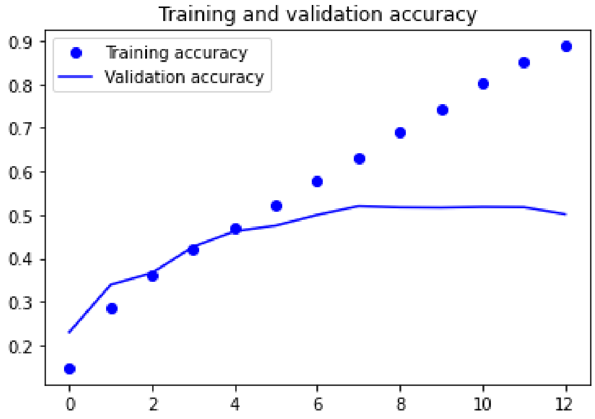
Appendices



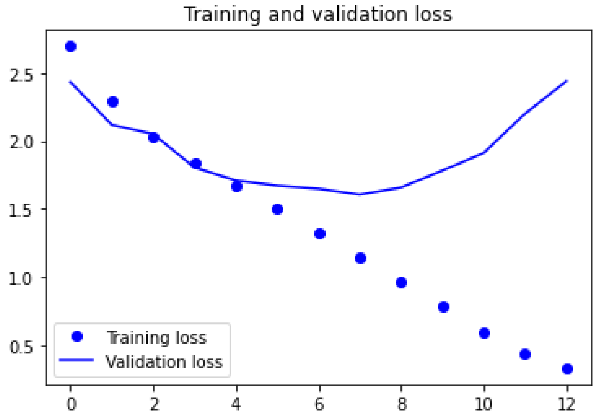
(a) Normal Images Without Dropout.



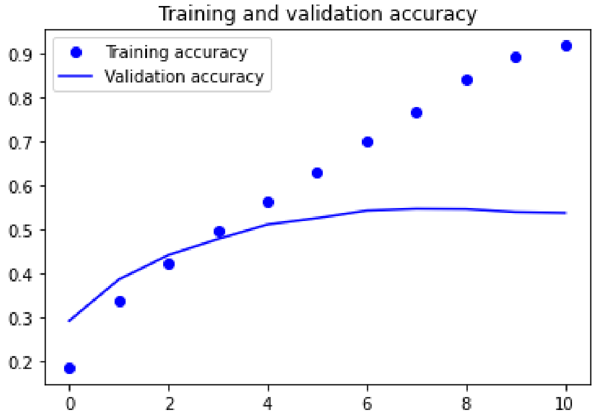
(b) Normal Images Without Dropout.



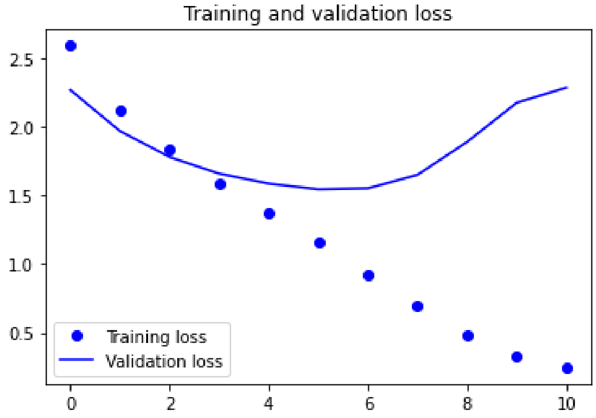
(a) Augmented Dataset Without Dropout.



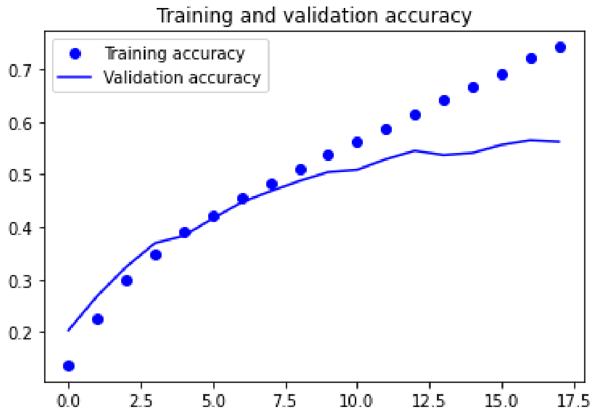
(b) Augmented Dataset Without Dropout.



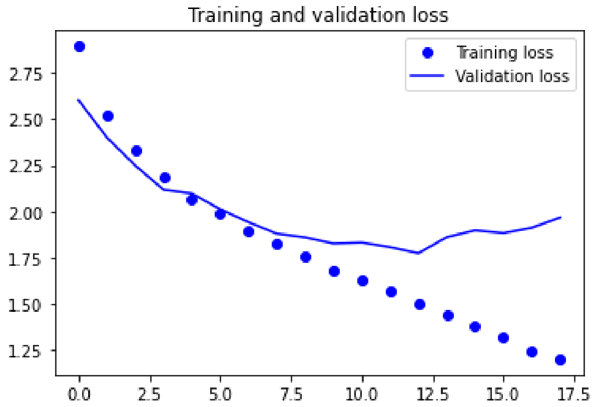
(a) Normal Images With SI Regularization Term Without Dropout.



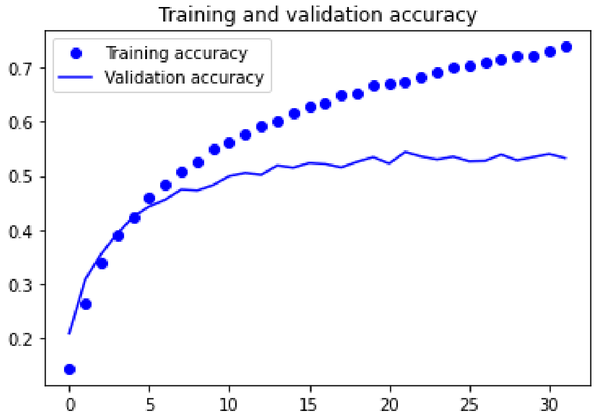
(b) Normal Images With SI Regularization Term Without Dropout.



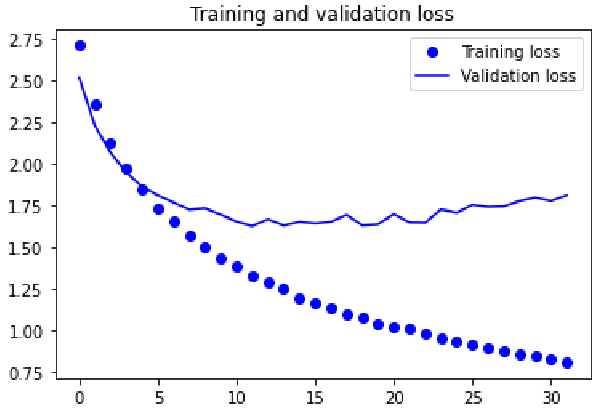
(a) Normal Images With L2 Regularization Without Dropout.



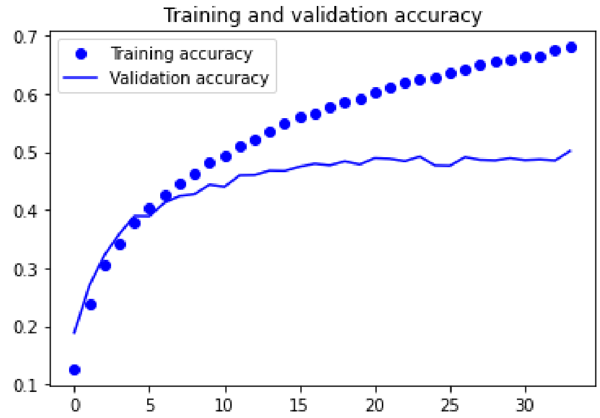
(b) Normal Images With L2 Regularization Without Dropout.



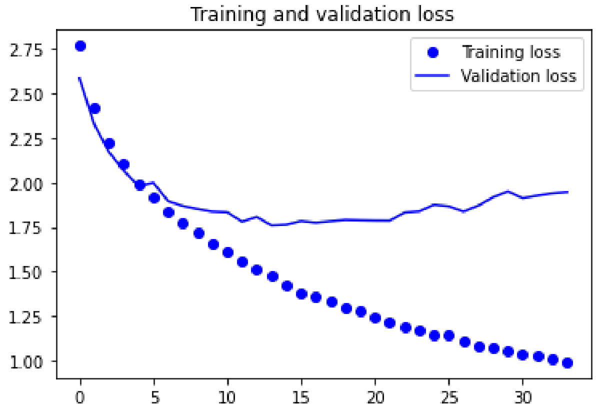
(a) Normal Images With Regular Dropout.



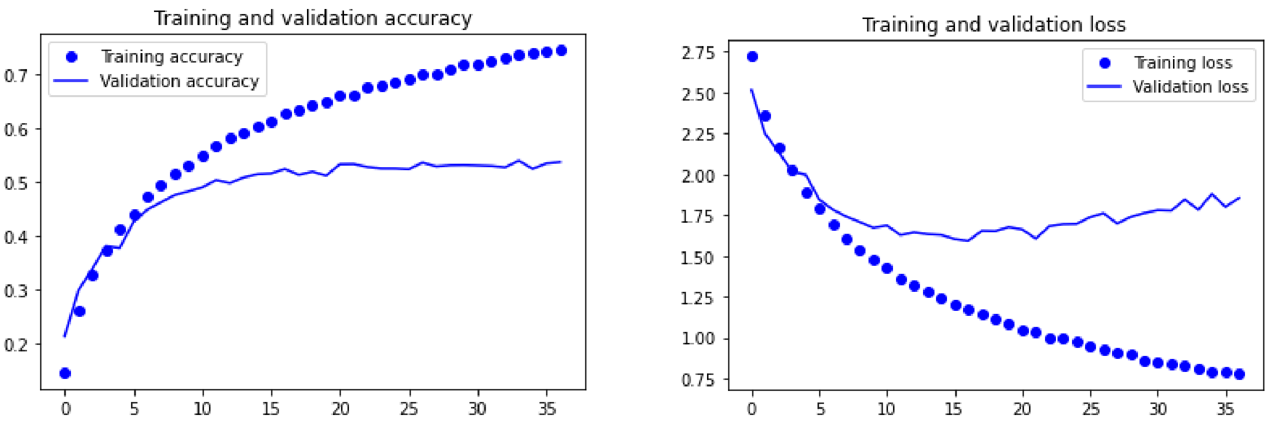
(b) Normal Images With Regular Dropout.



(a) Augmented Dataset With Regular Dropout.

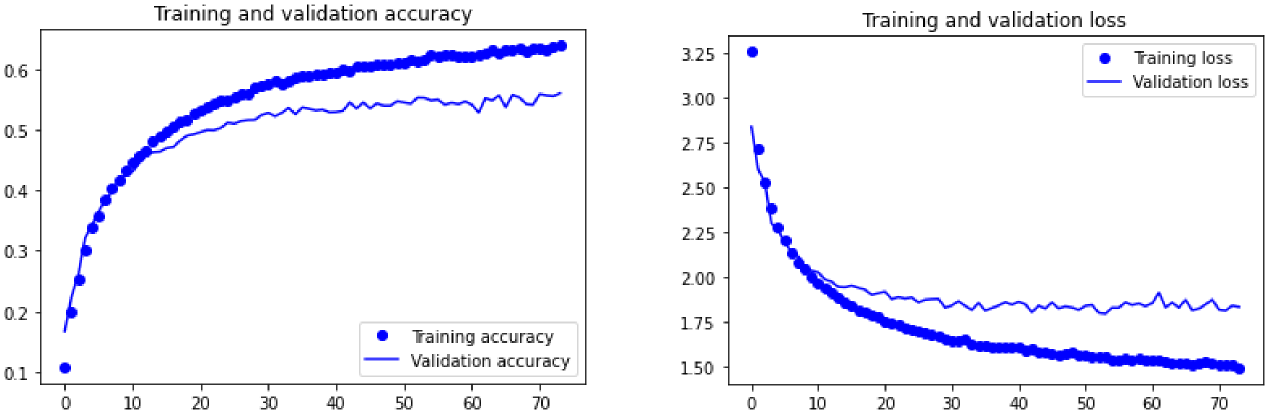


(b) Augmented Dataset With Regular Dropout.



(a) Normal Images With SI Regularization Term With Regular Dropout.

(b) Normal Images With SI Regularization Term With Regular Dropout.



(a) Normal Images With L2 Regularization With Regular Dropout.

(b) Normal Images With L2 Regularization With Regular Dropout.