

Natural Scene Images Classification

Group 2: Yifan Luo
dept. Structural Engineering, UCSD
La Jolla, CA
yil904@ucsd.edu

Pengkun Li
dept. Electrical Engineering, UCSD
La Jolla, CA
p3li@ucsd.edu
&
Sen Fu
dept. Electrical Engineering, UCSD
La Jolla, CA
sefu@ucsd.edu

Yu Shi
dept. Electrical Engineering, UCSD
La Jolla, CA
s4yu@ucsd.edu

Abstract—Convolutional Neural Network (CNN) has demonstrated promising performance in image classification tasks. In this project, we proposed several models to conduct natural scene images classification, which are random forest (RF), support vector machine (SVM), artificial neural network (ANN) and CNN. We mainly focus on the performance and structure of CNN, more specific, the layers components and parameters adjustment. By comparing accuracy and errors of results in different models, their performance could be evaluated. How CNN works better in the computer vision field can be learned from this work.

Index Terms—Image classification, machine learning, convolutional neural network

I. INTRODUCTION

With recent advancements in technology, scene images classification has been greatly developed with machine learning techniques. Image Classification problem, the task of assigning an input image one label from a fixed set of categories, is one of the core problems in Computer Vision that, despite its simplicity, has a large variety of practical applications [2]. In this project, 14,000 images of 6 categories, which are building, sea, glacier, mountain, street, forest, are trained as input, 3,000 images as testing set. We selected 4 different models, which are Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Network (ANN), and Convolutional Neural Network (CNN) to classify images categories. When using different models to train data, we collected errors and scores from each model. By adjusting different parameters of each model, we examine effects on classification performance. Comparing each model performance, we focus on developing a higher accuracy or lower errors. At last, we summarize advantages and disadvantages of each models, conclude the best model for image classification in this project and discuss the significant findings. All codes are uploaded to Github. [1]

II. RELATED WORK

Natural scene classification is an open problem in computer vision and has wide applications in both image and video indexing.

In 1998, Maron et al. [3] used Multiple-Instance learning to classify images of natural scenes, which is highly relevant to our topic. This learning method is a way of modeling

ambiguity in supervised learning examples. In this paper, they described a framework for learning scene category concepts that can be used for content-based image retrieval from large database.

In 2004, Matthew et al. [4] presented a framework to handle the classification error problem and apply it to the problem of semantic scene classification. Their paper was based on the fact that a natural scene may contain multiple objects such that the scene can be divided into multiple categories. Their method was demonstrated on the SVM classifier.

In 2006, Anna Bosch et al. [5] proposed a scene image classifier. In this paper, the image classifier can learn categories and their distributions in unlabeled training images using pLSA (probabilistic Latent Semantic Analysis), and then implement the distribution in testing images as a feature vector in a k-nearest neighbor scheme. This paper provides an efficient method of image classification, however, it is for unsupervised dataset and the image data we using are labeled.

In 2015, Mandar Dixit et al. [6] introduced the fisher vector (FV) embedding to recognize the categories of images. The FV can exploit the properties of the descriptors space to summarize the bag of semantics, which is a representation of scene image with help of convolutional neural network. This state-of-the-art method has a highest accuracy of 72.86% on MIT Indoor dataset.

In 2017, Gong Cheng et al. [7] review recent progress of remote sensing image scene classification. In this paper, they compared several remote sensing image scene classification methods: handcrafted-featured-based method, unsupervised-feature-learning-based methods, and deep-feature-learning-based method. In the deep-feature-learning-based method, they discussed the two widely used deep learning methods, which are stacked autoencoder and convolutional neural networks. As a result, the deep-learning-based CNN features have the highest accuracy.

III. DATASET AND FEATURES

A. Introduction of Dataset and Features

The dataset was provided by Intel [8] to achieve the image classification. It was divided into two parts, training set and

test set. Each dataset is composed of pictures. Training set contains 14034 pictures and test set contains 3000 pictures. For both datasets, there are six classes: mountain, seas, forest, building, glacier and street. Each class has the approximate same numbers of pictures. The RGB pictures are in 150 * 150 size.

B. Image Processing Method

- 1) *Image Preprocessing*: the normalization was used to transfer the number of each point to the range between 0 and 1.
- 2) *Feature Extraction*: Image classification requires a lot of details of the figure to achieve a higher accuracy. After several practices, the final model does not contain the feature extraction process.

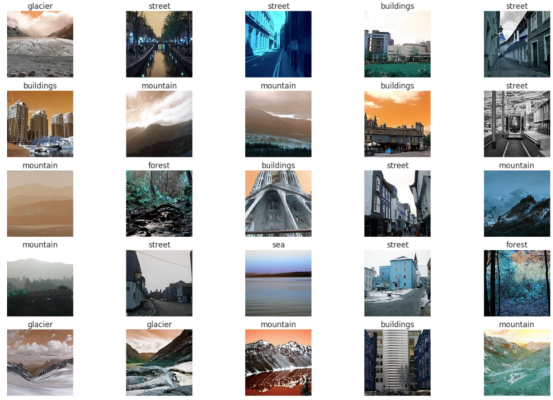


Fig. 1. Random 25 Sample Images of 6 classes

IV. METHODS

To compare and achieve the better result, four different models are used. They are Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest (RF) and Convolution Neural Network (CNN).

A. Support Vector Machine

Support Vector Machine (SVM) is a discriminative classifier used for classification and regression analysis. Examples are represented as points in space and a separating hyperplane is formally used to define it and classify them into different categories [9]. It is commonly used to perform non-probabilistic binary classification but we can extend its implementation to perform multi class classification. SVMs can not only perform linear classification but also nonlinear classification using kernel trick where their inputs are mapped into high-dimensional feature spaces. In Fig. 2, non-linearly separable 2-dimensional data points are mapped into 3-dimensional feature spaces with a hyperplane. SVMs perform best separation when the distance from the nearest data point on each side to the hyperplane or margin is maximized. Maximal margin effectively reduces the generalization error of the classifier.

In this project, we utilize soft-margin SVM to perform multi class and nonlinear classification using kernel trick. We try

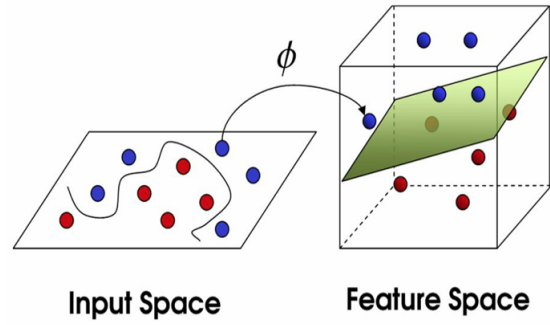


Fig. 2. Sample structure of Support Vector Machine

two types of kernels: Radial Basis Function (RBF) kernel and Polynomial kernel since they are used for non-linear hyperplane. Finally we choose RBF as our kernel because it has better performance on test data. RBF kernel is a kernel function commonly used in support vector machine which maps input data into high-dimensional feature spaces. As shown in equation (1), the RBF kernel of our SVM is the squared Euclidean distance between two feature vectors:

$$\mathcal{K}(x, x') = \exp(-\gamma \|x - x'\|^2) \quad (1)$$

Then we tune the parameters kernel, gamma and C of SVM since they have great effect on the performance of nonlinear SVM. The gamma parameter defines how far a single training example's influence reaches and the C parameter is a regularization parameter and the penalty for misclassified data which prevents overfitting. Finally we set gamma to 'scale' and C to 1.0 to optimize our model and improve the accuracy of it.

B. Artificial Neural Network

The artificial neural network (ANN) is based on a collection of connected units or neurons. The ANN can be used to approximate the relationship between input and output. [10] Since the ANN layers are fully connected and original images are 3-D pixel matrix, we need to preprocess the images at first. Each image is in RGB color with resolution of 150x150, so we need to flatten the 150x150x3 pixel matrix into a 1-D vector. In Fig.3, a sample pixel image with label "mountain" is read as three 150x150 matrix, reshaped into a 1*67500 image vector and then put into the ANN model for training.

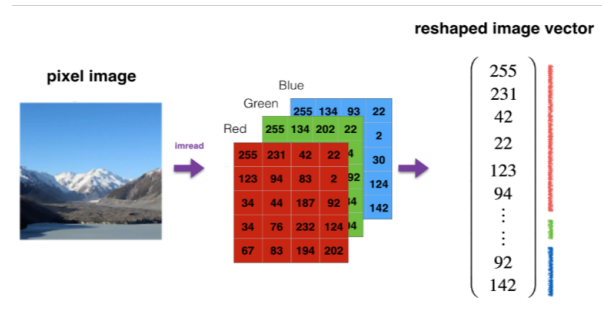


Fig. 3. Data preprocessing for a sample image

After the data preprocessing, we could construct the neural networks. The dimension of input vector is 1×67500 , the first hidden layer has a dimension of 1×128 , the other hidden layers were selected with dimension of 1×64 . Activation function for hidden layers is ReLU due to the benefits of sparsity and a reduced likelihood of vanishing gradient, for output layer is softmax function to turn the score produced by the neural networks into values that can be easily interpreted. We trained a relative shallow model at first with 3 hidden layers with dimension of 1×64 , then we train a deeper model with 22 hidden layers with dimension of 1×64 . When we increase the layer numbers, the accuracy of model will not increase significantly, which means this model has achieved its limitation.

Choosing sparse categorical cross entropy as the loss function, as shown in equation (2).

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C 1_{y_i \in C_c} \log p_{model}[y_i \in C_c] \quad (2)$$

The double sum is over the samples i , whose number is $N = 14034$ in training set, and the categories c , whose number is $C = 6$ in dataset. The term $1_{y_i \in C_c}$ is the indicator function of the i -th sample belonging to the c -th category. The $p_{model}[y_i \in C_c]$ is the probability predicted by the model for the i -th observation to belong to the c -th category. When there are more than two categories, the neural network outputs a vector of C probabilities, each giving the probability that the network input should be classified as belonging to the respective category.

Calculating the loss function L from equation (2), then conducted the updated weight vector and bias vector from equation (3).

$$\vec{W}_{n+1} = \vec{W}_n - \eta \frac{\partial L}{\partial W_n} \quad \text{and} \quad \vec{b}_{n+1} = \vec{b}_n - \eta \frac{\partial L}{\partial b_n} \quad (3)$$

In the fitting process, we chose "Adam" as our optimizer for its better performance than other gradient descent optimization algorithms [11], and selected the learning rate $\eta = 0.001$. The proportion of training and validation is 8:2, and the batch size is 512 for each batch. This batch size could improve the accuracy and speed of training.

C. Random forest

A random forest multi-way classifier consists of a number of trees, with each tree grown using some form of randomization. The leaf nodes of each tree are labeled by estimates of the posterior distribution over the image classes. Each internal node contains a test that best splits the space of data to be classified. An image is classified by sending it down every tree and aggregating the reached leaf distributions. Randomness can be injected at two points during training: in subsampling the training data so that each tree is grown using a different subset; and in selecting the node tests. [12]

Growing the trees. The trees here are binary and are constructed in a top-down manner. The binary test at each

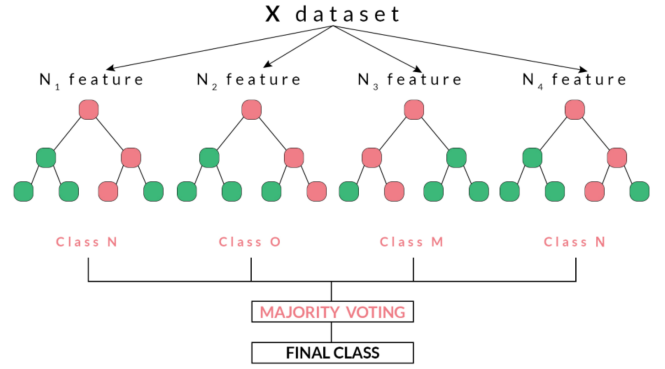


Fig. 4. Schematic diagram of Random Forest

node can be chosen in one of two ways: (i) randomly, i.e. data independent; or (ii) by a greedy algorithm which picks the test that best separates the given training examples. Best here is measured by the information gain

$$\Delta E = -\sum_i \frac{|Q_i|}{|Q|} E(Q_i) \quad (4)$$

caused by partitioning the set Q of examples into two subsets Q_i according to the given test. Here $E(q)$ is the entropy $\sum_{j=1}^N p_j \log_2(p_j)$ with p_j the proportion of examples in q belonging to class j , and $|\cdot|$ the size of the set. The process of selecting a test is repeated for each non-terminal node, using only the training examples falling in that node. The recursion is stopped when the node receives too few examples then it reaches a given depth.

Learning posteriors. Suppose that T is the set of all trees, C is the set of all classes and L is the set of all leaves for a given tree. During the training stage the posterior probabilities ($P_t(l(Y(I) = c))$) for each class $c \in C$ at each leaf node $l \in L$, are found for each tree $t \in T$. These probabilities are calculated as the ratio of the number of images I of class c that reach l to the total number of images that reach l . $Y(I)$ is the class-label c for image I .

Classification. The test image is passed down each random tree until it reaches a leaf node. All the posterior probabilities are then averaged and the arg max is taken as the classification of the input image. [13]

D. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a deep learning algorithm which is made up of neurons that have learnable weights and biases. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity. Our CNN model contains 15 layers. It is a deep network because the images are made up of complex features. In the architecture aspect, it is composed of convolutional layer, maxpooling layer and output layer.

For the convolutional layer, each neuron outputs a feature map to describe the input image. Six convolutional layers are introduced to the network. Firstly, a convolutional layer with

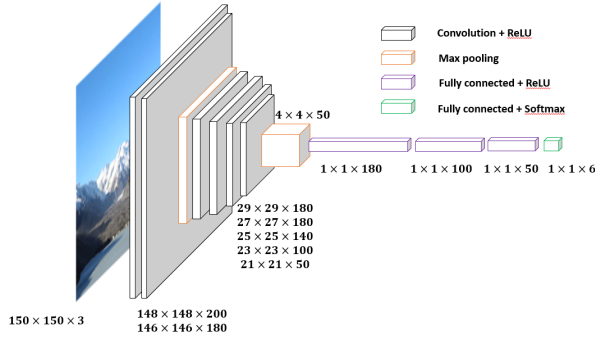


Fig. 5. CNN Architecture

200 channels is used to expand the number of feature maps then describe the image in more aspect to show the feature better. Then the number of channels continuously decreases as 200 → 180 → 180 → 140 → 100 → 50 → 50 to reduce the calculation complexity.

For the pooling part, two pooling layers are involved to reduce the spatial size as third and sixth layers. Pooling layer is not applied after each convolutional layer because the size of our input image is small. If too many pooling layers are used, a large information loss happens.

Four fully connected layers are introduced to decrease the length of image vector to 6 which represents 6 classes. The dimension of vector decreases steeply as 800 → 180 → 100 → 50 → 6. The reason is if the dimension of vector quickly drops from 800 to 6, information loss could increase the error rate.

Activation function for the last fully connected layer is softmax and ReLU for the remaining layers. ReLU function works for images as the inputs are non-negative values and also improves the sparsity. Softmax function could output a list of probabilities for corresponding classes.

Back propagation and sparse cross-entropy loss function are utilized to optimize. The definition of cross-entropy is showed in equation (2). "Adam" is the chosen optimizer because of the stable iteration performance.

V. RESULTS AND DISCUSSION

For each method discussed above, the training and testing error are listed in Table.1.

TABLE I
ERROR FOR EACH MODEL

Index	Model Name	Training Error	Testing Error
1	SVM	35.8%	50.4%
2	ANN	29.97%	43.17%
3	RF	0.072%	40.03%
4	CNN	7.00%	14.54%

A. Support Vector Machine (SVM)

In our SVM model, we set hyper-parameters kernel to 'rbf', C to 1.0 and gamma to 'scale' because 'rbf' is used for

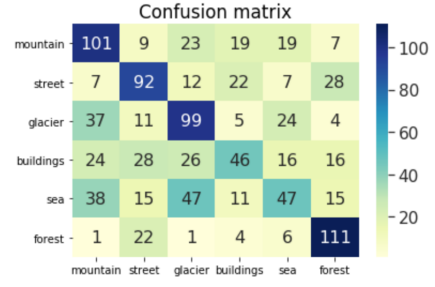


Fig. 6. Confusion Matrix of Support Vector Machine

nonlinear classification and 1.0 is a suitable value to prevent over-fitting. The primary metrics of our SVM model is the mean accuracy. The mean accuracy of our model on training data and test data are 0.642 and 0.496. Both accuracy is high because our test examples are highly nonlinear. Fig. 6 is the confusion matrix which visualizes the performance of our model on test data. It shows that although most images are classified correctly, some non-diagonal elements in the matrix still have high values. For example, the 'sea' and 'glacier' pair has value 47 and the 'street' and 'buildings' pair has value 28. The reason is that some 'street' and 'buildings' images look similar. Sometimes we can see buildings in an image labeled as 'street'. This makes the test data highly nonlinear and increases the test error of our SVM model.

B. Artificial Neural Network (ANN)

From the result of Support Vector Machine above, we find the accuracy is still need to be improved. Thus, we implement the Artificial Neural Network to pursue a better performance.

We trained 2 ANN models, the first model is relatively shallow with 3 hidden layers with 64 units as discussed above. But the error of the model achieved 85.43% in the testing set. Thus we developed our ANN model structure with more hidden layers to improve the classification ability. When the hidden layers of 64 units arrived 22, the ANN model have best performance with error of 42.73% in testing set.

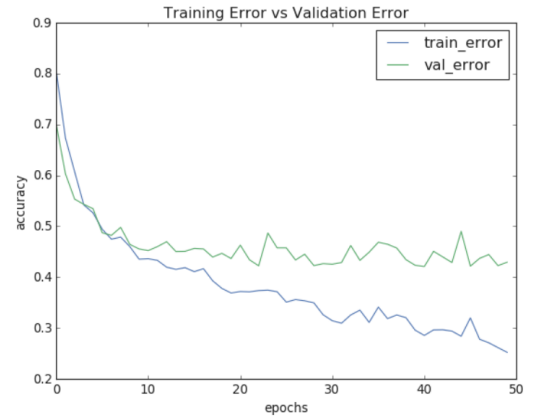


Fig. 7. Model Accuracy

C. Random Forest (RF)

When applying Random Forest classifier in this project, we try different hyper-parameters in RF. To start, we set 50, 100 and 150 n-estimators in our model. After that, mean squared error (MSE) can be calculated with respect to training data and test data. From result, we find that over-fitting problem appears. We get high accuracy in training data but low accuracy in test data. The mean squared error in testing is 3.197, whereas only 0.00584 in training data. When scoring this model by using model attribute of RF classifier, we get 1.0 score of test data. Meanwhile, experimental data shows that there is no rapid influence when only increasing the number of trees. Eventually, when the number of trees is 150, the best testing accuracy are gained. However, over-fitting is still the main problem of RF model. As follows, Fig. 8 gives confusion matrix of Random Forest Classifier.

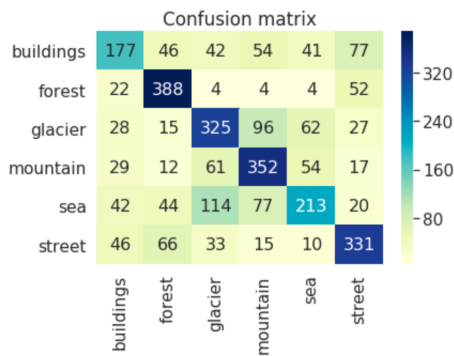


Fig. 8. Confusion Matrix of Random Forest

D. Convolutional Neural Network (CNN)

The main way to estimate the performance of CNN is the plot of error with epoch as Fig. 9. It shows the training error decreases from 61.12% to 7.00% and the test error decreases from 42.31% to 14.54%. The rate also decreases as epoch increases because of the convergence. At the start stage, the training error is higher than the test error because the training error is the average of the epoch and the test error is the error of the last moment.

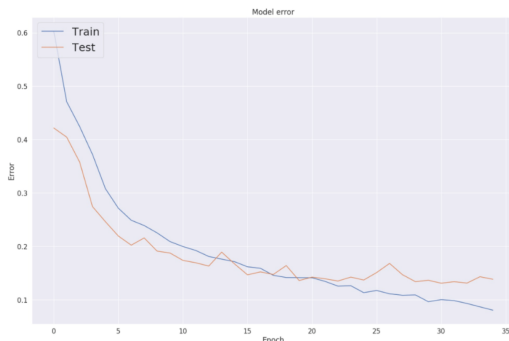


Fig. 9. Train and Test Error

Fig. 10 is the probability of corresponding image which is the output vector of CNN. The probability is not concentrated in one class. Three images have the positive probabilities in different class. Finally, the model will assign the image to the class with highest probability. It shows the tolerance of CNN model.

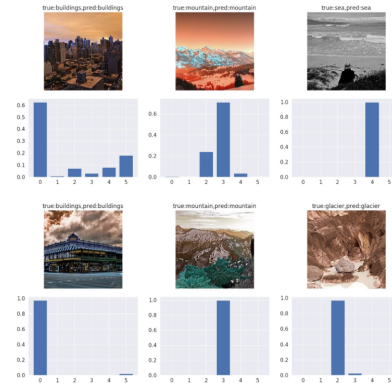


Fig. 10. Predicted Probability of 6 Classes

Fig. 11 is the confusion matrix which shows the number of images which assigned to different classes. The column is the predicted class and the row is the actual class. The confusion matrix has the largest number on the diagonal. It means the trained model could classify the image correctly in the most time. However, there are still some high values on the misclassified pairs. The highest one among the wrong results is the predicted "sea" and actual "glacier". This could be explained after observing the original figures, some glacier and sea pictures are similar to each other as they have the blue and white color and semblable texture.

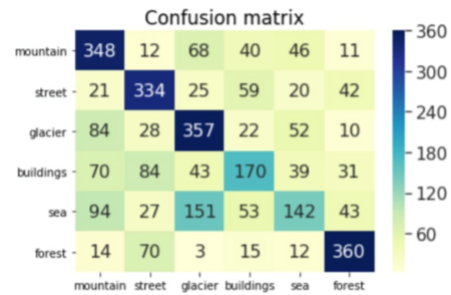


Fig. 11. Confusion Matrix

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

As shown above, four trained models can be evaluated and selected based on the accuracy of image classification. Supported by the excellent performance of feature extraction of CNN, we tried to adjust the parameters of CNN to achieve a lower error in both training and testing set.

In our expectation, the convolutional neural network should have the best accuracy or lowest error. However, in our first

attempt, the performance of CNN is worse than Random Forest. After the parameters adjustment, we overcome the problem of over-fitting which causes the higher error of CNN.

The final result with the selected CNN model error in testing set achieves 14%, which is higher than our goal with 10%. It is caused by the similarity among different categories, such as sea and glacier, or street and building. The error could be decreased by increasing the data size and construct a more advanced CNN structure.

B. Future Work

Feature extraction engineering will be introduced, in our works above, features are extracted by our model but not selected manually, which will improve the performance of model. Also, other pre-defined CNN models like VGG or AlexNet will be introduced for comparison.

REFERENCES

- [1] <https://github.com/jasonfu0516/ECE-228-Group2>
- [2] Chen, Chi-hau. *Handbook of pattern recognition and computer vision*. World Scientific, 2015.
- [3] Maron, Oded, and Aparna Lakshmi Ratan. "Multiple-Instance Learning for Natural Scene Classification." *ICML*. Vol. 98. 1998.
- [4] Boutell, Matthew R., et al. "Learning multi-label scene classification." *Pattern recognition* 37.9 (2004): 1757-1771.
- [5] Bosch, Anna, Andrew Zisserman, and Xavier Muoz. "Scene classification via pLSA." *European conference on computer vision*. Springer, Berlin, Heidelberg, 2006.
- [6] Dixit, Mandar, et al. "Scene classification with semantic fisher vectors." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.
- [7] Cheng, Gong, Junwei Han, and Xiaoqiang Lu. "Remote sensing image scene classification: Benchmark and state of the art." *Proceedings of the IEEE* 105.10 (2017): 1865-1883.
- [8] <https://www.kaggle.com/puneet6060/intel-image-classification>
- [9] Li, Xuchun, Lei Wang, and Eric Sung. "Multilabel SVM active learning for image classification." *2004 International Conference on Image Processing, 2004. ICIP'04.. Vol. 4. IEEE, 2004*.
- [10] Rebizant W. et. al. 2011. *Fundamentals of System Analysis and Synthesis. Digital Signal Processing in Power System Protection and Control Signals and Communication Technology*. 29-52.
- [11] Ruder, Sebastian. "An overview of gradient descent optimization algorithms." *arXiv preprint arXiv:1609.04747* (2016).
- [12] Anna Bosch, Andrew Zisserman, Xavier Munoz "Image Classification using Random Forests and Ferns" University of Girona
- [13] L. Breiman. *Random forests*. *Machine Learning*, 45:532, 2001.