Deep Learning for Cloud Detection

M. Le Goff [(1,2)], **J.-Y. Tourneret**[(1)], **H. Wendt**[(1)], **M. Ortner**[(2)], **and M. Spigai**[(2)]

[(1)] IRIT/ENSEEIHT/TéSA, University of Toulouse and [(2)] IRT Saint Exupery, Toulouse, France

Keywords: Deep learning, feature engineering, patch-based classification, superpixels

Abstract

The SPOT 6-7 satellite ground segment includes a systematic and automatic cloud detection step in order to feed a catalogue with a binary cloud mask and an appropriate confidence measure. However, current approaches for cloud detection, that are mostly based on machine learning and hand crafted features, have shown lack of robustness. In other tasks such as image recognition, deep learning methods have shown outstanding results outperforming many state-of-the-art methods. These methods are known to produce a powerful representation that can capture texture, shape and contextual information. This paper studies the potential of deep learning methods for cloud detection in order to achieve state-of-the-art performance. A comparison between deep learning methods used with classical handcrafted features and classical convolutional neural networks is performed for cloud detection. Experiments are conducted on a SPOT 6 image database with various landscapes and cloud coverage and show promising results.

1 Introduction

The generation of cloud masks associated with remote sensing images is an important issue in order to feed catalogues not only with images but also with cloud information. This problem has received considerable interest in the literature, cf., e.g., [1]. Upon request, the SPOT catalogue interface returns product meta-information, a cloud mask obtained from a semi-automatic pipeline, and a low resolution version of the requested image, called album version. This information gives insights on the image quality and can serve as a criteria for fast reprogrammation in case of too cloudy images. Contrary to popular object detection approaches where the detection consists in predicting bounding boxes, the pixel precision of the mask is important in order to remove cloud contaminated pixels from advanced processing.

Cloud detection consists of labeling each pixel of a scene by a binary variable indicating whether this pixel corresponds to a cloud or not. The labeling map is referred to as the cloud mask. The first cloud detectors were based on morphological operations such as shadow matching or on physical models specific to clouds (for example using dedicated spectral bands, see [2] for instance). However, these detectors were shown to lack generalization capabilities and robustness since they are satellite-dependent and can provide poor performance for specific images. The second type of cloud detectors is based on handcrafted features and machine learning ([1, 3, 4]). A set of handcrafted features is computed for each pixel and used by a machine learning algorithm to predict if a pixel belongs to a cloud or not. Unfortunately, this approach that currently achieves the best performance relies heavily on feature engineering to find the best features. In contrast, deep learning methods are thus appealing because they remove the feature engineering step from the model by learning its own features containing both spatial and spectral information.

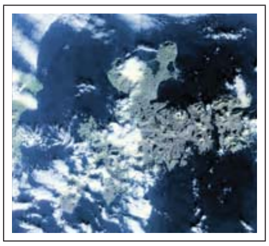

Figure 1. Example of an image with clouds.

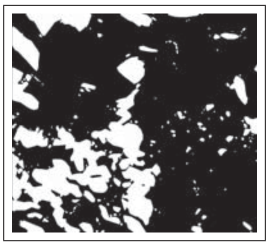

Figure 2. Associated cloud mask

The main idea of the cloud detection method investigated in this paper is to use a convolutional network operating on an input window to produce a label probability for each pixel. The convolutional net is fed with raw image pixels, and trained in supervised mode from fully-labeled images to produce a cloud index for each pixel. Convolutional networks are composed of multiple stages including a convolution module, a non-linearity, and a spatial pooling module. With end-to-end training, convolutional networks can automatically learn various hierarchical feature representations. Note that deep learning methods have provided amazing results in various image processing applications. These applications include image recognition [5] and semantic image labeling [6].

The new cloud detection method proposed in this paper is then compared to two traditional approaches using handcrafted features fed to a classic neural network or superpixels combined with a neural network for classification. The analysis of the obtained results allows one to appreciate the performance of each strategy for cloud detection. The data used in the experiment to evaluate the different input formats is a database of SPOT 6 album images associated with their cloud masks, which is presented in Section 3. Section 4 presents the different classification methods, including convolutional networks, that are used for cloud detection. Simulation results are presented in Section 5 whereas our conclusions are reported in Section 6. The results obtained in this work show that the features learned by the proposed convolutional neural network outperforms classical handcrafted features.

2 Related Work

As stated before, classical cloud detectors based on machine need a set of handcrafted features computed for each pixel in order to predict if the pixel belongs to a cloud or not. The main factor of detection performance is the choice of the handcrafted features [3].

Two main families of features have mainly been studied for this problem: spectral-based features and texture-based features [3]. The spectral features, such as raw spectal band values or more complex combinations such as differences or ratios between bands, have proved to be effective [2] but usually fail to distinguish between some objects such as ice and cloud which have similar behaviour in the spectral domain. Spectral features are also known to be highly sensitive to detector noise or atmospheric effects. In contrast, texture-based features are less sensitive to these effects. Texture methods are mostly based on the spatial distribution of numeric counts, either estimated with statistical measures such as the grey level cooccurence matrix [7] or with the help of known filters such as Gabor or discrete cosines [8].

The use of the parallax feature introduced in [1] has been proven to be highly effective but its computation requires to have access to the panchromatic (PAN) image which is not always possible when dealing with album images. The parallax feature compares the registration of the different color bands to the panchromatic in order to estimate the elevation map in the image. The elevation image can easily be used to recover clouds.

The main machine learning techniques that are used to learn the decision function from these features are support vector machines and neural networks ([3]). Unfortunately, none of these method (combination of features and classifier) have met the criteria of both accuracy and robustness. Performance improvement can be obtained by looking for more efficient features combinations and classifier. Deep learning are thus appealing because they learn both the features and the decision function at the same time for the classification. The first experiments on deep learning applied to cloud detection have shown promising results [4].

Cloud Coverage	Percentage of images
0%-1% - (A)	34.3%
1% - 10% (B)	15.1%
10 % - 25% (C)	11.6%
25 % - 75% (D)	18.9%
75 % - 100% (E)	20%

Table 1. Repartition of the cloud coverages in images.

3 Images and features

Album SPOT 6 images consist of 4-channel images acquired in the blue, green, red and near infrared wavelength domains. The spatial resolution of these album images is significantly smaller than those of the full resolution images in order to reduce memory requirements, while the radiometric resolution is preserved at 12 bits. A database of more than 10000 SPOT 6 album images, containing a large and representative variety of cloud coverages and landscapes, has been provided by Airbus Defense and Space. The images have been corrected for radial distorsion, internal sensor geometry and radiometric distortion. Our ground truth will be the cloud mask drawn by an operator. However, the cloud boundaries can be subject to controversy because of the noise presence due the variability during the segmentation by the operator. Note that this noise effect can have a strong impact on the final classification performance.

3.1 Features

Classical machine learning methods use features or numerical descriptors to perform classification. Image features can be computed from the four channels of the album images for all the image pixels. The objective of this work is to compare the pixel raw accuracy obtained with different classification methods. We will try to compare the performance of four types of handcrafted features.

- RGBI raw pixel values
- the corresponding band ratios (i.e., the ratio of the image intensities of two channels)
- Gabor coefficients
- Discrete cosine transform coefficients

As stated before, the choice of these features is critical to achieve a good classification performance: for instance, the raw pixel values are known to be highly sensitive to noise and classifiers based on raw values usually lack of robustness. Significant progress has been achieved in the design of handcrafted features for many image processing applications. These features usually include neighboring information and also physical correction. Band ratios are commonly used as remote sensing features and have shown great results for cloud detection. To include scale information, this paper proposes to compute band-ratios at three different spatial resolutions (60m, 120m and 240m). A pixel is then described by a total of 18 features.

Gabor features have been widely used for encoding textural properties of images. As stated in Section 2, they have also been successfully used for detecting clouds: Gabor features are computed for 4 different angles and 3 different scales which makes a total of 48 features per pixel. Discrete cosine transform is also known to provide an efficient encoding of the image. The DCT coefficients are also computed at 3 different scales with a block size of 4 which makes a total of 192 features.

3.2 Superpixels

Neighboring pixels of remote sensing images can have lots of similarities. As a consequence, classification could also be performed at a region level, i.e., groups of similar pixels are gathered together to form regions. The main advantage of this kind of approach is to gather statistics for similar pixels in order to make a reliable decision. Thus, group level statistics are much less affected by the presence of noise than pixel level statistics. In our experiment, superpixels are extracted using the SLIC superpixel algorithm [9], which iteratively creates regions using a k-means algorithm. Cross-validation led us to divide each image into 250 homogeneous regions (this number was fixed a priori and was not part of the optimization algorithm). 3D histograms associated with the distribution of the pixel values inside a region were used as descriptors for learning. A label indicating to which region each pixel belongs is finally assigned to each image pixel. Note that this method strongly relies on the segmentation method used for detecting the image regions.

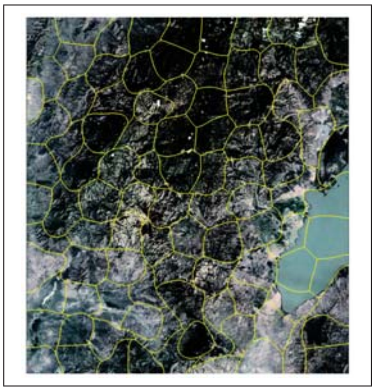

Figure 3. Example of SLIC superpixel segmentation on a tile of an image. Boundaries between superpixels are marked as yellow lines.

3.3 Patches

The usual learning approach is known to be limited by the feature engineering step, which can be critical and limit classification performance. Conversely, features learnt by convolutional networks (CNNs) are optimized for the classification task, which significantly improves classification accuracy. CNNs usually require a fixed input size. The experiments conducted in this paper correspond to 32×32 patches. The patch size has been chosen to fit both performance, memory and computing time requirements. Note that the patches investigated in this work correspond to 1 kilometer large area on the ground. The extracted patches are finally fed to the CNN. A patch centered on a given pixel will be used as input to the CNN for predicting the class of this pixel. The predicted class is then compared to the class of the pixel extracted from the cloud mask.

Note that any post processing consisting of smoothing the predicted mask or combining predictions is out of the scope of this paper. Indeed, the objective of this work is to compare the pixel raw accuracy obtained with different classification methods.

4 Neural Networks for cloud detection

4.1 Notations

Denote as $\mathcal{S} = \{(\boldsymbol{x}_i, y_i), i = 1, ..., N\}$ the training set containing N feature vectors $\boldsymbol{x}_i \in \mathbb{R}^d$ and their corresponding labels y_i ($y_i = 1$ means that the pixel belongs to the cloud class and $y_i = 0$ otherwise) used for a classical machine learning problem. The goal of a supervised machine learning algorithm is to build a classifier $f : \mathbb{R}^d \mapsto [0, 1]^2$ from a (possibly infinite) set of classifiers $\mathcal{F}$ that minimizes the training loss defined as

$$\epsilon\,(\mathcal{S}, f) = \sum_{i=1}^{N} L(y_i, f(\boldsymbol{x}_i)) \tag{1}$$

where L is a given loss function. In this paper, we will use the cross entropy loss function defined as $L(y_i, f(\boldsymbol{x}_i)) = y_i \log(f(\boldsymbol{x}_i)) + (1 - y_i) \log(1 - f(\boldsymbol{x}_i))$, which is commonly used for neural networks.

The decision function f is usually parametrized using a set of weights $\boldsymbol{W}$. These weights are optimised for minimizing the training loss function described above. Fortunately, the objective function is of the form $E(\boldsymbol{W}) = \frac{1}{N}\sum_{i=1}^{N} E_i(\boldsymbol{W})$, the weights can be learned using the stochastic gradient descent method (see [10] for more details). The individual weight $\boldsymbol{w}$ can be updated as $\boldsymbol{w} := \boldsymbol{w} - \eta \nabla_{\boldsymbol{w}} E_i(\boldsymbol{W})$, where η is the learning rate and i is randomly sampled between 1 and N. Training examples are then shuffled and fed to the network for updates. Slightly different updates such as [11] are used here to stabilize and speed up the optimization.

4.2 Convolutional neural networks

Convolutional networks [12] aka CNNs are a specialized kind of neural networks processing grid-like data including images. These networks have been used successfully in many applications. The goal of a CNN is to extract hierarchical features from the input image trough convolutions. Recent research has studied many new solution for faster and more reliable learning, including the rectified linear unit. ReLU is a neuron with a simplified non-linearity which allows much faster training, and over-fitting reduction. The first CNN exploiting all these solutions, proposed in [5], improved image classification results by more than 10% w.r.t. the previous state of the art. Current CNN architectures are composed of several layers, of various types:

Convolution layer A convolution takes as a first argument an input matrix I of size $M \times N \times B$, as a second argument a kernel K and outputs the following matrix $F = I * K$ (where $*$ denotes the convolution) known as the feature map

$$F(i,j,b) = \sum_k \sum_m \sum_n I(m,n,k) K_b(i-m, j-n, k).$$

One advantage of a CNN is referred to as "parameter sharing" meaning that the weights are used multiple times contrary to the fully connected layer. Note that the content of $K = (K_b)_{b=1,\dots,B}$ contains free parameters to learn during training.

Activation layer A nonlinear activation layer is usually applied after each convolution or fully connected layer. A nonlinear mapping such as $x \mapsto \max(0,x)$ is usually applied elementwise, i.e., $\boldsymbol{h} = \max(0, \boldsymbol{s})$ and called activation function

Pooling layer The pooling function replaces the output of the net at a given location by a quantity summarizing its neighborhood. The max pooling of a CNN usually computes the maximum value over a rectangular neighborhood. Pooling operations are invariant to small translations and deformations.

Fully connected layer Simple neural networks consists of stacking layers with hidden units. The output of a layer is computed using the linear product between the input and its weight matrix $\boldsymbol{W}$ of size $p \times q$, where p is the number of output units and q is the number of input units such that $\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x}$. An example of a simple neural network with 100 hidden units performing a binary classification from a 18 dimensional input vector (using band ratio features) is defined by $f(\boldsymbol{x}) = bW_2 \times \max(0, \boldsymbol{W}_1\boldsymbol{x})$, where $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are 100×18 and 2×100 matrices.

An artificial neural net (ANN) is obtained by concatenating more layers or more hidden units than this simple example. Contrary to all hidden layers, the output layer generally does not contain any activation function, especially if we want that the output layer provides the class scores of the classification rule. Neural networks are generally characterized by their size, which corresponds to the number of parameters to learn. For example, in the example above, the network has $100 \times 18 + 2 \times 100 = 2000$ parameters to learn.

A convolution layer needs a fixed grid-structured input, explaining why patches (of size $32 \times 32 \times 4$) of the input image are fed to the network. The network can thus learn using appropriate features computed for each patch for cloud detection. The convolutional network used in this work is composed of two convolution layers whose output matrix is flattened to a 1-D vector and fed to a fully connected layer. This structure corresponds to a classical CNN architecture.

5 Experiments

The main goal of this paper is to compare the performance of a ConvNet architecture applied to patches with the one obtained with a simple neural net with classical handcrafted features. We compare 6 different classifiers: an ANN applied to raw pixel values, an ANN applied to ratio features, an ANN applied to Gabor features, an ANN applied to DCT features, an ANN applied to superpixels and a CNN architecture applied to patches. The different classifiers are learnt using the same training set of images and evaluated with the same test dataset. The training and test datasets are constructed from images chosen randomly in the initial dataset described in Section 3. In practice, a training and test set used for this experiment are composed of more than 1 million pixels/overlapping patches. Features are computed from the values of the image pixels in the four channels. Superpixels are extracted using the SLIC algorithm as stated in Section 3.2. The pixel distribution is then computed and sent to the corresponding ANN. Finally, image patches are extracted and fed to the last network. Not that optimizing the network hyperparameters such as the number of regions, the overlapping ratio of patches, ... is out of scope of this paper. The CNN hyperparameters were adjusted using similar values than the network used for CIFAR-10 competition [13]. This network has shown great performance for classifying multiple images whose size is close to the ones considered in our use case. The network and its parameters are displayed in Fig. 5.

5.1 Evaluation metrics

The performance of each network is evaluated by the pixel difference between the generated cloud mask and its estimation referred to as pixel accuracy. Even if the accuracy is the main criterion in the comparison between the different approaches, the recall and precision for each classifier are computed. As a reminder, the recall is the proportion of non cloud pixels in the set of pixels predicted as clouds and the precision is the proportion of pixels predicted as clouds in the set of cloud pixels. Given the imperfection around boundaries of clouds in the operator cloud mask, the attention should be focused on the precision: the error on the boundaries are less important than the presence or not of a cloud. Note that the classification performance could also be evaluated using the percentage of clouds de-

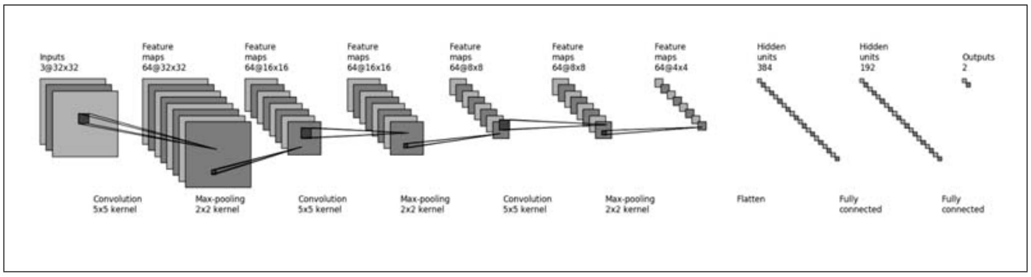

Figure 4. CNN used for classifying clouds. Structure is simlar than the one used for the CIFAR-10 competition

Input Type	Network	Accuracy	Recall	Precision
Gabor	ANN	77%	43%	66%
Raw Pixels	ANN	83%	68%	77%
Features	ANN	81%	68%	80%
Superpixels	ANN	83%	69%	80%
DCT	ANN	85%	75%	80%
Patches	CNN	86%	75%	81%

Table 2. Pixel accuracy for the different networks. Both learning and testing sets are composed of 500 images and the network hyperparameters of each network have been optimized using cross-validation.

Input Type	Complexity
Raw Pixels	×1
Ratio and Gabor	×1
DCT and CNN	×2
Superpixels	×4, 8

Table 3. Number of weights to be optimized by net compared to the pixel raw net. The superpixel net has almost five times more weights to optimize than the raw net.

tected in each image to remove the effects of boundaries in the evaluation metric. However, this metric requires an object detection step, which highly depends on the algorithm used.

5.2 Results

The pixel accuracies displayed in Table 2 show the performance associated with each of the proposed methods. For the handcrafted features, the raw features and Gabor features lead to the worst performance. The use of ratios of band intensities and the multiscale approach improves the performance compared to the raw values and the DCT features provides the best accuracy for the handcrafted features. The use of superpixels also leads to one of the best accuracy. Although, it should be mentioned that according to Table 3, it is associated with one of the biggest networks in terms of number of weights.

An important result is that all the handcrafted features are outperformed by the learnt features of the CNN applied on patches. The CNN provides the best accuracy, which is slightly better than the DCT features. Note that the extracted features of the CNN are optimized for cloud detection because of its end-to-end training whereas the DCT features are intended to be used for general purposes including image compression. Note also that the structure of the DCT classifier and the CNN are similar and composed of a feature extraction step with convolutions and a decision function mainly based on two fully connected layers. Moreover, according to Table 3, adding parameters to learn new features with convolutions (as done in the CNN) does not add more weights compared to the DCT ANN classifier. Thanks to parameter sharing, the majority of weights are gathered in the fully connected layers.

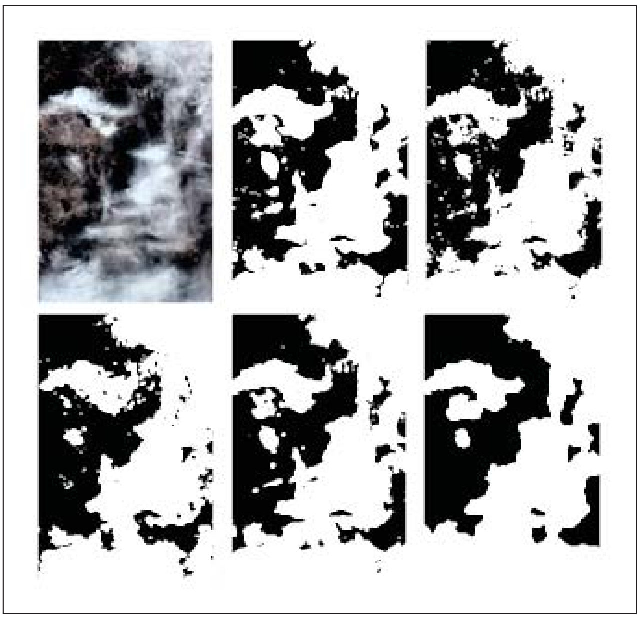

Figure 5. Predicted images for each network. From right to left and up to down (a) Input Image (b) Operator mask (c) Raw pixel prediction (d) Feature pixel prediction (e) Patch prediction (f) Superpixel prediction

To conclude, Figure 5 shows that from a visual point of view the superpixel and patch detectors have to be preferred because they are much less noisy than the pixel detectors. However, the superpixel detector tends to miss small clouds which is mostly caused by the size of regions extracted from the image.

6 Conclusion

Deep learning offers the possibility to build really complex and robust classifiers. With the rise of new technologies such as cloud computing and Graphical Processing Units, convolutional networks are faster to train. The problem addressed in this paper was to investigate the use of convolutional networks for cloud detection and to compare the resulting classification performance with state-of-the-art algorithms. We compared different neural network architectures with different input vectors. The performance of the different neural networks was also evaluated on an industrial database of SPOT 6 album images. We showed that the patches with convolutional networks is clearly the best solution for our experimental settings. This choice allows the classifier to learn meaningful features about the spectral content and shape attributes of clouds for classification. A second important result is that the use of superpixels in ANN improves classification performance when compared to pixel detectors. Future work includes the study of convolutional networks exploiting semantic segmentation [6] of images. Note that this kind of structure needs some structure adaptation to be used for remote sensing images. Indeed, remote sensing images are usually much larger than common images and their size is highly variable.

Acknowledgements

The study was conducted in the frame of the OCE contract managed by IRT Saint Exupery, Toulouse (France) and sponsored by the following funding partners: Agence Nationale de la Recherche, Airbus Defence and Space, Thales Alenia Space, Airbus Defence and Space Geoinformation, Telespazio, Airbus Group Innovation, LAAS-CNRS, IRIT.

References

[1] Latry C., Panem C., and Dejean P., "Cloud detection with SVM technique," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS'07)*, Barcelona, Spain, 2007, pp. 448–451.

[2] Jedlovec G., "Automated detection of clouds in satellite imagery," 2010.

[3] Geethu Chandran A. and Christy J., "A survey of cloud detection techniques for satellite images," 2015.

[4] Shi M., Xie F., Zi Y., and Yin J., "Cloud detection of remote sensing images by deep learning," in *Geoscience and Remote Sensing Symposium (IGARSS), 2016 IEEE International*. IEEE, 2016, pp. 701–704.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NIPS'12)*, Lake Tahoe, Nevada,US, 2012, pp. 1097–1105.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *arXiv preprint arXiv:1412.7062*, 2014.

[7] Tian B., Shaikh M. A., Azimi-Sadjadi M. R., Haar T. V. H., and Reinke D. L., "A study of cloud classification with neural networks using spectral and textural features," *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 138–151, 1999.

[8] Onsi M. and ElSaban H., "Spatial cloud detection and retrieval system for satellite images," *International Journal of Advanced Computer Science and Applications, (IJACSA)*, vol. 3, no. 12, 2012.

[9] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 34, no. 11, pp. 2274–2282, 2012.

[10] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*, pp. 421–436. Springer, 2012.

[11] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[12] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 1995, 1995.

[13] Krizhevsky A. and Hinton G., "Learning multiple layers of features from tiny images," 2009.