32nd European Modeling & Simulation Symposium
17th International Multidisciplinary Modeling & Simulation Multiconference

ISSN 2724-0029 ISBN 978-88-85741-44-7 © 2020 The Authors.
DOI: 10.46354/i3m.2020.emss.021

Comparing Classification Methods for the Building Plan Components

Daniel Wilfing[1] and Oliver Krauss[2*]

[1]VIEW Promotion GmbH, Frauscherberg 8, Friedburg 5211, AUSTRIA
[2]Research Group for Advanced Information Systems and Technology (AIST), Research and Development Department, University of Applied Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg, AUSTRIA

[*]Corresponding author. Email address: oliver.krauss@fh-hagenberg.at

Abstract

The classification methods Histogram of Oriented Gradients, Bag of Features, Support Vector Machines and Neural Networks are evaluated to find a fitting solution for the automatic classification of building plan components. These components feature shapes with little features and minor differences. After processing the building plans for the classification, feature analysis methods, as well as machine learning based approaches, are tested. The results of the classification methods are compared and the behaviors of the classification methods are analyzed. First results have shown, that neural network classification using line data extracted via Hough transformation and additional calculations surpass other classification methods tested in this work. It was found that the basic structure of building plan components can be detected with neural networks, but further improvements have to be made, if only a single classification process is to be relied on. In the future this work will be used to create 3D building models from 2D plans and enable agent based simulation in the models.

Keywords: Classification; Modeling & Simulation; machine learning; feature analysis; building plan

1. Introduction

Building plans contain different components, which define how rooms are set up and connected to form an apartment, house or building complex. These components clearly differentiate the different parts in the building and are defined by a set of standards. However, different standards exists depending on state or even county legislature. The components can be combined in a multitude of ways on the building plan, and architects may draw the building plans by hand, leading to minor differences for the same components. These differences regarding standards and drawings are no problem for human observers, but lead to problems for computers utilizing detection methods.

Currently, the modeling process for buildings is done by the architect or a specialized designer, usually only for bigger projects or for specific requests, as the modelling process is too costly for small scale or older projects (Wonka et al., 2003). An automated analysis of building plans would make the modeling process quicker and more affordable. The 3D-models can be used for different calculations and simulations, for example automatically calculating the shortest escape route or verifying the accessibility of the building. Additional analyses, can improve the design process of new building plans, as well as enabling the use of simulations for existing building plans.

Classification methods can solve this problem and detect the different components of the building plan correctly. A multitude of different classification methods exist, and it's difficult to predict, which methods

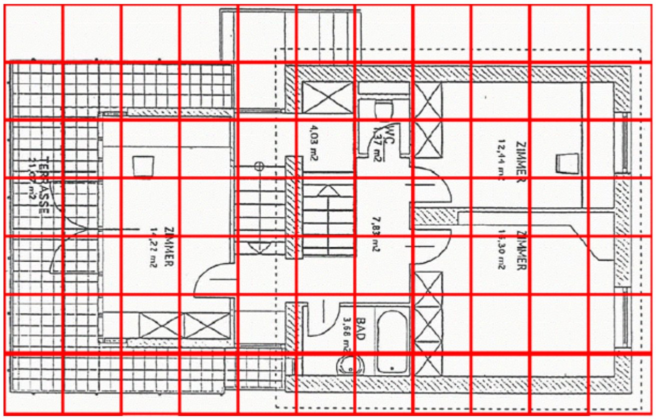

Figure 1. Image-based classification preprocessing - Building plans are split into squares. Each square is then independently classified.

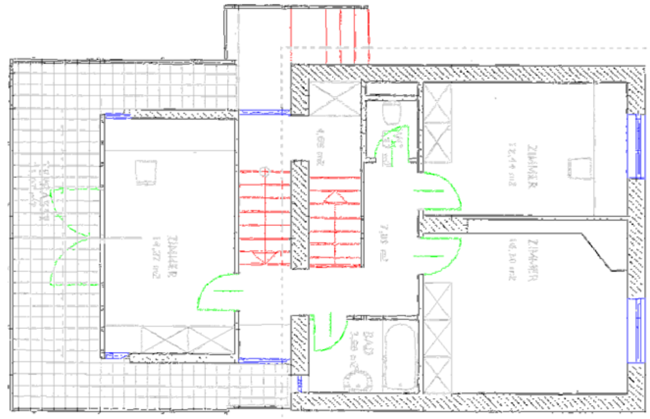

Figure 2. Five different components (*walls* - black, *doors* - green, *windows* -blue, *stairs* -red and *unknown* -grey) were classified manually.

are suitable for the given use case. Additionally, preprocessing of the building plans has to be done depending on the classification. Some classification methods work with images from the building plan, while others require additional information calculated based on the building plans (Lu and Weng, 2007).

2. Methods

Classification methods analyze objects and categorize them according to different criteria. Some classifiers are able to calculate values for the criteria themselves, while other classification methods require a set information and adapt fitting parameter sets during a training phase. Classification methods, like histogram of oriented gradients (HOG) and bag of features (BOF), are able to classify images without requiring specific preprocessing or calculations (Nowak et al., 2006; Dalal and Triggs, 2005). Classifications using machine learning methods on the other hand are not immediately able to classify images, but require a transformation of the image data into a set of values.

In this work, the image classification algorithms are tested using cropped parts of the building plan. Tests are done with squared image parts, as well as extracted lines with a padding. For the machine learning classifications, the lines composing the building plan are extracted and information about them is calculated, like length, angle and number of neighboring lines. The calculated values can then be used to train the classifiers, thus allowing to classify new lines, and in further consequence entire components of the building plan.

2.1. Preprocessing for Image-Based Classification

Preprocessing of the building plans is required to prepare the images for the classification. To improve the component detection, the building plan is binarized. The building plan is first transformed to grayscale, after which a threshold was chosen for binarization using a histogram (Sauvola and Pietikäinen, 2000). This process removes noise in the background, which could otherwise be wrongly classified as a component.

The image based classification methods require larger images than the machine learning based classification to identify features from. Building plans are split into same sized parts, which are then evaluated. An example for splitting the building plan like this can be seen in Figure 1. Squares the size of 50x50, 75x75 and 100x100 pixels are tested and evaluated. These parts are manually classified, before the classifiers are trained. Additionally, a second method is tested. Instead of splitting the building plan into squares, the building plan will be fragmented into its lines, including a padding for each line. Regarding the padding, tests will be done with padding sized 50, 75 and 100 pixels.

2.2. Histogram of Oriented Gradients

In this classification process, the shape of an image is described by calculating a histogram of gradient directions for smaller regions of the image. These regions are called cells. Histogram of gradient directions are calculated for each cell (Dalal and Triggs, 2005). Due to the usage of cells, geometric transformations don't influence the classification process, with the exception of rotation. By comparing the histograms of the different cells, it is possible to classify the images (Luo et al., 2015). Histogram of oriented gradients is tested using cropped, equally sized parts of the building plan. Different image sizes will be tested to verify, which sizes lead to the better results. The cell sizes 4x4, 8x8 and 16x16 will be tested. A test phase with images of the extracted lines is done as well and compared to the previous results.

2.3. Bag of Features

This classification method calculates a number of different features, which are then quantized, to find how certain features define specific classes (Csurka et al., 2004). Often calculated features are textures, and other repeating patterns in the image. New images can then be classified by calculating clusters for the different features (Fei-Fei and Perona, 2005). Similarly to the test phase with histogram of oriented gradients, bag of features classification is tested with cropped images, as well as extracted lines with padding. The same padding sizes (50, 75 and 100 pixels) will be tested.

2.4. Preprocessing for Machine Learning Classification

Background noise is removed via binarization. After this process, the lines of the building plan can be extracted. Hough transformation has been implemented for tracking the start- and end-position of lines (Hough, 1962). The different lines composing the building plan can be extracted and are classified by coloring them. This coloring and classification process is done manually, and is required for the training phases of classifiers. Figure 2 depicts such a manually classified building plan. Using this information, the classifiers can deduce which features describe a specific class.

The machine learning classification methods require a transformation from the image data to set of values. For this, the lines calculated in the previous steps are used. For each line a set of properties is calculated, like length, angle and number of neighboring lines. To further improve the classification, more properties are beneficial (Bishop, 1995). This is solved by adding the properties of the closest neighboring lines to the current line, doubling the number of properties for a single line. A single line then contains enough properties for a stable classification.

2.5. Support Vector Machines

Support vector machines classify objects with vector space calculations. During the training phase, the objects are placed in a vector space. A line is calculated, which is able to separate the objects by class (Cortes and Vapnik, 1995). If the line separates the different objects with a large margin classification improves. Without separation, the vector space is transformed in a higher dimensional space and the line calculation is repeated (Jin and Wang, 2012).

This classification method does not work directly with images, but requires a set of values. The properties of the line data will be incorporated for the training and test phase of the support vector machine classifier. Different classifier settings are evaluated, like different binary learners and boosting algorithms (Hsu and Lin, 2002).

2.6. Neural Networks

This method creates a network of nodes, which are connected to each other via edges. Each edge has a weight assigned, defining how calculations are propagated through the network (Kolen and Kremer, 2001). During the training of the neural network, edges and weights are adjusted, until the network is able to classify the given value sets accurately (Ojha et al., 2017). The neural network approach is evaluated using the value sets containing properties of all lines. Neural networks with 50, 75, 100, 500, 1000 and 1500 neurons and a depth of 1, 2, 5 and 10 are evaluated.

3. Results

Confusion matrices show the results over 15 plans from 7 different architects for the tested classification methods.

Square size: 50 x 50
Cell size: 4 x 4

	Uknown	Wall	Window	Door	Stairs
Unknown	**94.00%**	5.00%	1.00%	0.00%	0.00%
Wall	22.00%	**77.00%**	1.00%	0.00%	0.00%
Window	35.00%	26.00%	**39.00%**	0.00%	0.00%
Door	**43.00%**	16.00%	3.00%	38.00%	0.00%
Stairs	32.00%	27.00%	0.00%	0.00%	**41.00%**

Average of correctly classified: **57.80%**

Square size: 75 x 75
Cell size: 8 x 8

	Uknown	Wall	Window	Door	Stairs
Unknown	**92.00%**	5.00%	1.00%	1.00%	1.00%
Wall	19.00%	**77.00%**	1.00%	2.00%	1.00%
Window	**39.00%**	25.00%	36.00%	0.00%	0.00%
Door	**41.00%**	16.00%	2.00%	39.00%	2.00%
Stairs	38.00%	0.00%	8.00%	0.00%	**54.00%**

Average of correctly classified: **59.60%**

Square size: 100 x 100
Cell size: 8 x 8

	Uknown	Wall	Window	Door	Stairs
Unknown	**85.00%**	10.00%	3.00%	2.00%	0.00%
Wall	21.00%	**71.00%**	6.00%	0.00%	2.00%
Window	27.00%	22.00%	**49.00%**	0.00%	2.00%
Door	36.00%	23.00%	3.00%	**38.00%**	0.00%
Stairs	11.00%	11.00%	21.00%	5.00%	**53.00%**

Average of correctly classified: **59.20%**

Figure 3. Success rates for the HOG Feature Selection using cropped squares. Numbers shown bold and underlined define which class has been correctly classified the most often for a specific class.

Padding size: 50 x 50 Cell size: 8 x 8	Uknown	Wall	Window	Door	Stairs
Unknown	**92.00%**	5.00%	1.00%	1.00%	1.00%
Wall	17.00%	**80.00%**	1.00%	1.00%	1.00%
Window	34.00%	17.00%	**46.00%**	2.00%	1.00%
Door	38.00%	6.00%	1.00%	**52.00%**	3.00%
Stairs	27.00%	14.00%	1.00%	1.00%	**57.00%**
Average of correctly classified:					65.40%

Padding size: 75 x 75 Cell size: 8 x 8	Uknown	Wall	Window	Door	Stairs
Unknown	**84.00%**	11.00%	1.00%	2.00%	2.00%
Wall	26.00%	**70.00%**	1.00%	1.00%	2.00%
Window	**42.00%**	17.00%	38.00%	3.00%	1.00%
Door	38.00%	13.00%	1.00%	**46.00%**	1.00%
Stairs	39.00%	11.00%	0.00%	1.00%	**49.00%**
Average of correctly classified:					57.40%

Padding size: 100 x 100 Cell size: 8 x 8	Uknown	Wall	Window	Door	Stairs
Unknown	**85.00%**	10.00%	1.00%	1.00%	3.00%
Wall	27.00%	**70.00%**	1.00%	1.00%	1.00%
Window	**42.00%**	15.00%	39.00%	1.00%	3.00%
Door	39.00%	11.00%	1.00%	**47.00%**	2.00%
Stairs	35.00%	9.00%	1.00%	2.00%	**53.00%**
Average of correctly classified:					58.80%

Figure 4. HOG using padded lines are similar to those of cropped squares, with the exception of 50x50 padded lines and a cell size 8x8, which lead to 7.3% better results.

3.1. Histogram of Oriented Gradients (HOG)

This classification method was tested in two test phases using different image data. The first test phase was run with evenly cropped squares of building plans. An average classification rate of 58.87% is reached, with *unknown* and *wall* classes being correctly classified the most consistently. The cell size of 8x8 lead to the best results in this test phase, except for the squares sized 50x50, where better results could be achieved with a cell size of 4x4. The results of the best runs for the different square sizes can be seen in Figure 3.

The classifier strongly preferred components of the class *unknown*. This could be, because this class is the most prevalent in building plans (see the grey lines in Figure 2). The shape of images class *unknown* also strongly varies, while the shapes of other classes differ less. This could lead to *unknown* being chosen more often.

Similar results have been observed for the test phase using extracted lines with additional padding. The average classification rate was 60.53% and the cell size of 8x8 lead to better results than 4x4 and 16x16. The best result with 65.40% was achieved with a padding size of 50x50 and a cell size of 4, which can be seen in Figure 4. Classifying the extracted line images lead to similar problems as using the cropped square images. Components of the class *unknown* are strongly preferred, which might be thanks to their diverse shapes, as well as their prevalence in the test data. Splitting this class up into other classes could fix this issue, as more shapes would then be distributed over more classes, instead of the *unknown* class.

3.2. Bag of Features

Bag of features performed the worst compared to the other classification methods in this work. The calculation of the required features was not possible during the classifier training. Without any informative features, the classification process cannot be done successfully, thus rendering any following tests with the classifier futile. This behavior was noticed during the test phase, as every object was classified as *unknown*, which was the most common class type during the test phase. This behavior was observed for both the cropped squares, as well as the extracted lines, of the building plans. The size of the squares and padding of the lines didn't lead to any changes regarding this result.

The reason for the failure of this classification method could be, that features are mainly created by analyzing textures. If no textures are available, or the given textures are not diverse enough, then the classification process is worsened (Lazebnik et al., 2006).

3.3. Support Vector Machines

The classification results of support vector machines are decent for some of the components, but fail to classify the other components. *Unkown* and *wall* components have been classified correctly more consistently than the other classes, which can be seen in Figure 5. Wrong classification as class type *unkown* was especially prevalent. The best results with support vector machines have been achieved via a "One Vs One" binary learner and the boosting algorithm "Gentle Boost", which, on average, correctly classified 1.38% more objects than the run with "One Vs All" and 4.62% more objects than run with "Ada Boost M1" (Friedman et al., 2000).

All three support vector machine settings resulted in heavily preferring *unkown* to the other classes. A reason for this could be, that the information describing the class *unkown* is so diverse, that the clustering methods couldn't split the different classes well enough. The results could be improved by splitting the class *unkown* up in additional classes. More distinct clusters might then be found, which will also improve the classification of the previous components.

Binary learner: One Vs All — Boosting Algorithm: Gentle Boost

	Unknown	Wall	Window	Door	Stairs
Unknown	83.18%	12.43%	1.44%	2.05%	0.90%
Wall	35.78%	56.77%	3.84%	1.97%	1.65%
Window	48.14%	28.15%	19.58%	2.65%	1.48%
Door	62.11%	16.62%	2.89%	15.69%	2.69%
Stairs	44.59%	19.39%	3.83%	4.74%	27.45%

Average of correctly classified: 40.53%

Binary learner: One Vs One — Boosting Algorithm: Gentle Boost

	Unknown	Wall	Window	Door	Stairs
Unknown	83.50%	12.06%	1.40%	2.08%	0.96%
Wall	35.51%	57.30%	3.86%	1.91%	1.42%
Window	47.31%	27.49%	21.26%	2.46%	1.47%
Door	60.86%	16.39%	2.86%	16.97%	2.91%
Stairs	42.89%	18.61%	3.49%	4.47%	30.55%

Average of correctly classified: 41.92%

Binary learner: One Vs One — Boosting Algorithm: Ada Boost M1

	Unknown	Wall	Window	Door	Stairs
Unknown	81.33%	13.38%	1.55%	2.28%	1.45%
Wall	39.07%	52.23%	4.34%	2.11%	2.24%
Window	51.20%	28.11%	16.03%	2.95%	1.73%
Door	63.97%	15.60%	3.55%	13.92%	2.97%
Stairs	45.72%	22.27%	4.19%	4.86%	22.97%

Average of correctly classified: 37.29%

Figure 5. With Support Vector Machines lines of the class *unknown* have been correctly classified most often, but other classes have also been wrongly classified as this class most of the time

3.4. Neural Networks

Neural networks have been found to lead to the best results (72.82%) of all presented classification methods in this work (see Figure 6). Components of the class *unkown*, *wall* and *window* are correctly classified at least 74% of the time, while *door* and *stair* components are recognized 63% to 75% of the time. This classification, is able to classify the other components as well. Because of this, the detection of *unkown* is worse compared to the other classification methods in this work (9.69% worse than support vector machines and 14.02% to 17.35% worse than histogram of oriented gradients, depending on the method).

The classification rate for *wall* components is a 17.55% better than the classification rate of support vector machine, but 0.35% to 2.02% worse than the results achieved using histogram of oriented gradients. *Window*, *door* and *stair* components are significantly better classified by the neural network approach, leading to 17.14%, up to 54.03%, better classification results.

Different neural network sizes have been evaluated, ranging from 50 to 1500 neurons. Good results have been achieved starting with a size of 75. Bigger neural networks improved the results slightly, but didn't lead to any significant improvements (Lawrence et al., 1996).

Different amounts of hidden layers have been evaluated as well, with the best results being achieved using 2 hidden layers with 100 neurons each. Neural networks with 5 hidden layers and between 50 and 100 neurons lead to good results as well. Adding more hidden layers to the network worsened the results significantly, as the classifier tried to focus on mainly the *unkown* and *wall* classes, ignoring components of other classes. This behavior seems contradictory, as more hidden layers with more neurons would allow more complex calculations, thus should be able to incorporate all compo-

Neurons: 100 — Hidden Layer: 2

	Unknown	Wall	Window	Door	Stairs
Unknown	80.79%	10.50%	2.34%	4.39%	1.97%
Wall	14.57%	78.84%	2.44%	2.54%	1.61%
Window	12.51%	9.17%	75.45%	1.76%	1.11%
Door	18.33%	10.19%	1.94%	65.84%	3.70%
Stairs	13.16%	7.17%	2.13%	2.94%	74.59%

Average of correctly classified: 75.10%

Neurons: 1000 — Hidden Layer: 1

	Unknown	Wall	Window	Door	Stairs
Unknown	75.72%	13.24%	2.78%	5.51%	2.75%
Wall	16.44%	75.59%	3.31%	2.45%	2.21%
Window	12.85%	11.42%	73.23%	1.33%	1.17%
Door	17.17%	11.93%	2.90%	64.77%	3.23%
Stairs	15.87%	9.90%	3.68%	3.36%	67.19%

Average of correctly classified: 71.30%

Neurons: 1500 — Hidden Layer: 1

	Unknown	Wall	Window	Door	Stairs
Unknown	76.45%	12.99%	2.81%	5.04%	2.71%
Wall	16.32%	76.46%	2.83%	2.52%	1.87%
Window	13.62%	11.47%	71.22%	1.20%	2.49%
Door	19.66%	12.40%	1.71%	63.54%	2.69%
Stairs	14.43%	8.54%	2.85%	3.53%	70.66%

Average of correctly classified: 71.67%

Figure 6. In Neural Networks, different sizes of networks only lead to minor differences, but a minimum size is required to allow more complex classification calculations.

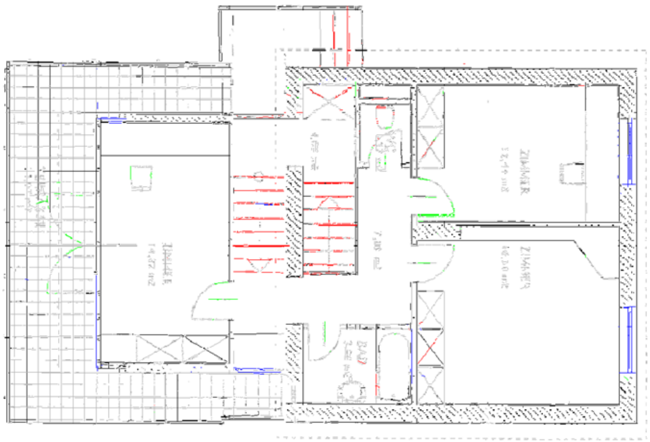

Figure 7. Result of the neural network classification of one building plan, showing that the core elements are classified correctly.

nents in the classification process. However, this could be because of overfitting and overtraining, as bigger networks are able to specialize on the different values of the *unkown* and *wall* classes (Lawrence et al., 1996). Overfitting may occur to smaller scaled neural networks as well, but the amount of data given for the building plan component analysis is too large, leading to a more shallow classification, which proves beneficial when classifying a dataset with unevenly distributed classes. Additionally, smaller networks can make use of different methods to further prevent overfitting, which are not available to large networks (Hinton and Salakhutdinov, 2006).

Figure 7 shows a building plan classified using a neural network. The basic structure of the building was correctly classified. Not everything was classified correctly, especially components of the class *door* (green color in Figure 7) could only be classified partially. Other times similar classes have been mistaken with each other, like *stair* and *wall* components. This could either be fixed by further improving the neural network classification, or implementing additional algorithms for fixing possible mistakes made during the classification. One possible solution would be through a rule set, defining which line and component combinations are possible. Wrong classifications, like a *wall* passing through a *stair* or a *window* missing a supporting *wall*, could be recognized and reevaluated. Another option would be to combine the neural network with other classification methods, or additional neural networks, to create ensembles. Classifiers could be found, which are able to classify specific components more precisely, which could then be combined into an ensemble to improve the overall classification performance over multiple classes (Opitz and Shavlik, 1996).

4. Conclusion

Out of four different classification methods, only one was able to consistently classify all different building plan components. Most methods are able to correctly classify at least two components, but are not able to consistently classify the remaining ones. Bag of features was not able to calculate the necessary features, rendering the following classification process moot. Histogram of oriented gradients and support vector machine prefer one or two specific classes and are not able to differentiate the other classes well enough. Neural network classification is able to differentiate those minor differences and allows a consistent classification of building plan components.

Neural network classification is able to correctly classify a majority of the lines and at least a few lines for each component. The basic structure of the building is correctly classified, which would allow for an automatic transformation into a 3D-model, as well as further calculations, like escape route and thermal calculations. Improving the neural network methods via additional algorithms or ensembles could further improve the classification, allowing the detection of more intricate building plan structures.

5. Outlook

To improve the work tests will be done with neural network classification. Different settings will be evaluated and additional classes will be added, as well as new parameters to classify. Ensembles of neural networks will be evaluated as well to determine, if one large network, or several smaller specialized networks, lead to better classification results for building plan components. Different methods will be evaluated to prevent overfitting and overtraining in larger networks, like dropout or prevention of co-adaption (Srivastava et al., 2014; Hinton et al., 2012). Additional building plan data will be incorporated into a next test phase as well, to improve the classifier training and provide new component information. Especially building plans of artistic or historic buildings containing unusual structures will be evaluated, to determine the limit of the neural network classification. The goal of this work is to use the classification to create a building model out of the 2D plan and translate it into a 3D model to allow different evaluations, such as escape route planning or simulation of evacuations. Next steps will also involve rule sets to combine the correctly detected lines into building plan components.

6. Funding

This work was funded by the Austrian Research Promotion Agency (FFG), and was conducted in Cooperation with Aicher IT Consulting GmbH.

References

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273–297.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893 vol. 1.

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 524–531 vol. 2.

Friedman, J., Hastie, T., and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *Ann. Statist.*, 28(2):337–407.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580.

Hough, P. V. C. (1962). Method and means for recognizing complex patterns.

Hsu, C.-W. and Lin, C.-J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425.

Jin, C. and Wang, L. (2012). Dimensionality dependent pac-bayes margin bound. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1034–1042, USA. Curran Associates Inc.

Kolen, J. F. and Kremer, S. C. (2001). *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies*. IEEE.

Lawrence, S., Giles, C. L., and Tsoi, A. C. (1996). What size neural network gives optimal generalization? convergence properties of backpropagation. Technical report, UniversityofMaryland.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2169–2178.

Lu, D. and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870.

Luo, Z., Chen, J., Takiguchi, T., and Ariki, Y. (2015). Rotation-invariant histograms of oriented gradients for local patch robust representation. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 196–199.

Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In Leonardis, A., Bischof, H., and Pinz, A., editors, *Computer Vision – ECCV 2006*, pages 490–503, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ojha, V. K., Abraham, A., and Snásel, V. (2017). Metaheuristic design of feedforward neural networks: A review of two decades of research. *CoRR*, abs/1705.05584.

Opitz, D. W. and Shavlik, J. W. (1996). Generating accurate and diverse members of a neural-network ensemble. In *Advances in Neural Information Processing Systems*, pages 535–541. MIT Press.

Sauvola, J. and Pietikäinen, M. (2000). Adaptive document image binarization. *Pattern Recognition*, 33(2):225 – 236.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Wonka, P., Wimmer, M., Sillion, F., and Ribarsky, W. (2003). Instant architecture. *ACM Trans. Graph.*, 22(3):669–677.