Automated Image-Based Detection of State of Construction Progress

Hesam Hamledari, E-mail: hesamh@stanford.edu
Department of Civil and Environmental Engineering, Stanford University

***Abstract*—Construction projects usually experience schedule and cost overruns. To avoid such dire consequences, it is imperative that construction progress be regularly monitored. This work presents an automated technique for the visual detection of construction progress in images, using machine learning and digital images. A cascade scheme is employed which is primarily based on the use of bag-of-visual-word and texture classification techniques. The method receives as input an image of an indoor under-construction partition, and it automatically classifies it based on its state of progress. The algorithm was observed to perform robustly at indoor sites, obtaining an average classification accuracy rate of 96%.**

1. INTRODUCTION

Smart construction monitoring is essential to a project's on-schedule and on-budget completion [1]. In recent years, there has been a great momentum toward using machine learning and digital images for automated monitoring of construction equipment [2,3,4], materials [5,6,7], workers [8, 9], and state of progress [10, 11]. However, monitoring of indoor construction, and in particular indoor partitions (walls) still requires more attention [12, 13,14].

This work introduces an automated machine learning-based technique which receives as input an image of an indoor wall, and it automatically detects its state of progress. In other words, the technique classifies input images based on their state of progress (framing, insulation, installed sheets, plastered sheets, and painted wall). The technique employs bag-of-visual-word and texture classification techniques in a cascade scheme. Through multi-stage support vector machine-based binary classifications, images are categorized into their corresponding state of progress. The proposed solution was tested on 750 digital images of actual site conditions, and it reached an average 96% accuracy rate.

2. RELATED WORKS

The detection of construction progress for indoor partitions is still in its early stages [12]. This is due to the highly cluttered and occluded scenes at indoor sites, extreme illumination conditions, and varying viewpoints [14].

Early efforts resulted in the development of an algorithm which the image's response to different filters such as Sobel as input to a learning algorithm. This machine learning-based technique could detect progress of indoor partitions for a *limited* number of states [12]. This work hugely relied on manual parameter tuning and required user's input with respect to approximate state of progress. Later works addressed the automated visual detection of construction objects at indoor sites including steel studs and insulation blankets [14, 15]. While these works can provide meaningful data on installed objects, they cannot robustly detect the progress of work and indoor scenes due to their low reliance on machine learning, need for manual tuning, and sensitivity to changes in the scene [15].

Hence, there is a need for techniques that can employ machine learning technique for robust detection of state of progress at indoor sites. The application of machine learning can help overcome challenges posed by dynamic nature of construction sites. While construction automation community sees the need for such studies on indoor progress tracking, studies with similar interest have been well studied in the computer science domain.

Scene classification in images [16], for example, has been a subject of interest for many years. Many studies have focused on the use of bag-of-visual-word techniques [16, 17, 18], first introduced by [19]. These techniques construct a visual vocabulary, also known as codebook, which can be used to encode images into a histogram based on the frequency of distinctive features inside the image. Other studies have used object recognition results as features for scene classification [20]. Due to their robustness to changes in the scene, such techniques can be beneficial to the indoor progress tracking problems.

Hence, this work emphasizes on the use of a machine learning-based technique for robust classification of scenes (images captured of partitions) at indoor construction sites.

3. DATA COLLECTION

For the purposed of this project, multiple visits to construction sites were planned, and a set of 750 color images (1920×1080 pixels) were captured by the author using a smartphone. These images all depict indoor under-construction partitions (walls), and they have been captured at different illumination conditions, viewpoints, and varying distances to the wall. Some samples of this dataset are illustrated in Fig. 1, categorized based on the depicted walls' state of progress.

All images in the dataset were manually labeled by the author (in terms of state of progress) for use in the algorithm's testing and training. Table 1 summarizes the distribution of images based on the wall's state of progress.

Figure 1. Examples of the collected images, categorized based on their state of progress

Table 1. The manually collected and labeled dataset of 750 digital images

Framing	Insulation	Installed	Plastered	Painted
150	150	150	150	150

4. METHODS

First, the problem definition and the overall design of the proposed method are discussed in sections 4.1 and 4.2. Then, the details with respect to the feature selection and learning techniques will be elaborated in sections 4.3 to 4.4.

4.1. Problem Definition

The proposed method receives as input an unlabeled digital image of an under-construction indoor partition, and it should automatically classify it into one of the following 5 states of progress (Fig. 1): 1) framing completed; 2) insulation placed; 3) installed drywall; 4) plastered drywall; and 5) painted drywall. For the sake of simplicity, we will refer to these states as *framing*, *insulation*, *installed*, *plastered*, and *painted*. The work on a partition first begins by constructing its frame, then insulation is placed within the wall's framing. Drywall sheets are installed on both sides of frame, and the sheets are plastered and then painted.

The classification of these five states benefits construction practitioners and various trades such as drywallers, painters, and framers. It is assumed that the method is not provided with any information other than the input image.

4.2. Overall Design

In this work, a combination of bag-of-visual-word (BOVW) and texture classification techniques are employed in a cascade scheme (Fig. 2). The proposed method receives an image as input and classifies it into one of the five states of progress. While other variations of this design were evaluated and will be discussed in section 5, this cascade scheme resulted in the highest performance. As shown in Fig. 2, the method solves 4 binary classification problems.

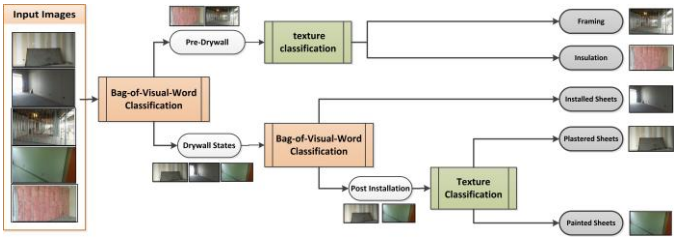

Figure 2. The proposed technique for image-based progress detection; input images are automatically categorized into one of the five states of progress: framing, insulation, installed sheets, plastered sheets, and painted sheets.

In the first stage, an SVM classifier categorizes images into two groups of "pre-drywall" and "drywall". The former contains images of both states of *framing* and *insulation*, and the latter contains images of the last three states. The first classification stage is performed using BOVW technique, the details of which will be provided in section 4.3.

In the second classification stage, images in "pre-drywall" group are further classified into two states of *framing* and *insulation*. This is achieved by using a material recognition algorithm that automatically detects insulation blankets inside an image, taking advantage of both color and texture. In the third stage, images in the "drywall" group are further classified into "installed" and "post-installation". The latter contains images in both *plastered* and *painted* states. This again relies on the use of a BOVW technique (section 4.3). In the last stage, a texture classification algorithm is developed to categorize the "post installation" images into two states of *plastered* and *painted*.

Through these 4 binary classification stages and the combined use of BOVW and texture classification algorithms, images are classified into the desired five states of progress. In the following sections, we introduce the BOVW technique (section 4.3) and texture and material classification algorithms (section 4.4).

4.3. Bag-of-Visual-Word Technique

The BOVW technique [19] plays an essential role in the method presented herein. In the following sections, we discuss the feature generation process and the learning technique used in the BOVW pipeline (Fig. 3).

4.3.1. Features

In this technique, local keypoints are first extracted from a set of images (e.g., training dataset), and then they are passed to a k-means clustering algorithm where each resulting cluster center represents a visual word, or a distinctive feature of images in the dataset. The resulting cluster centers form a visual vocabulary, also known as codebook (Fig. 3).

In this work, speeded up robust features (SURF) [21] keypoints are used for codebook generation. SURF has been inspired by scale invariant feature transform (SIFT) [22], and it is claimed to outperform it both in terms of run time and robustness to image transformations [23]. To detect keypoints,

SURF uses integral image for fast approximation of the determinant of the Hessian matrix H(p,σ):

$$H(p,\sigma) = \begin{pmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{pmatrix}$$

L(p,σ) are the grayscale image's second-order derivatives.

To acquire the descriptor, a square neighborhood is formed around the detected keypoints, and the sum of Haar wavelet [24] responses are calculated in its four sub-regions. This computation is also expedited using the integral image. In this work, a second variation of the SURF algorithm, namely dense SURF (DSURF) [25,26], is also evaluated (section 5). DSURF performs dense sampling on a regular dense grid (dividing the image into a series of patches). DSURF has been shown to outperform SURF in some research works [26].

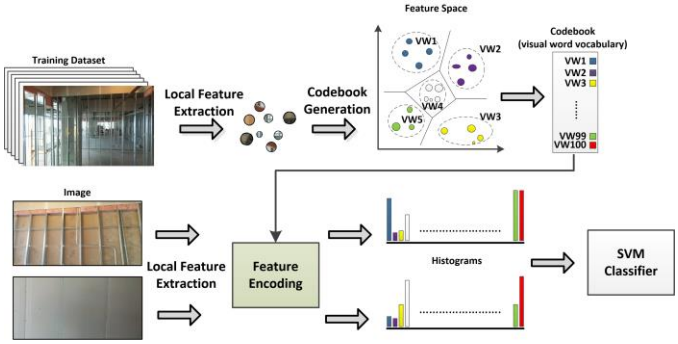

Figure 3. The bag-of-visual-word technique

To generate features used in the learning algorithm, a histogram should be created of the visual words observed in an image. In other words, the frequency at which each of the visual words in the codebook appears in an image is recorded, and the resulting histogram is used as input feature for the learning process. To encode an image into a histogram, local features (e.g., SURF or DSURF) are first detected, and then each of them are compared with visual words in the codebook and assigned to the closest one [28]. For example, if X is a series of D-dimensional descriptors, i.e., $X=[x_1, x_2,\ldots x_M] \in R^{DxM}$, and the codebook has N visual words, i.e. $Y=[y_1, y_2,\ldots y_N] \in R^{DxN}$, the encoding process finds a mapping function from X to Y. Some of the techniques that can be used for this purpose include vector quantization (VQ) [19], soft assignment encoding [29], sparse encoding [30], and Fisher kernel encoding [31]. In this work, VQ technique, also known as hard-assignment coding, is used where each feature descriptor is assigned to the closest visual word using l2-normalized Euclidean distance. If u_{mn} represents the mapping function from the x_m (m^{th} feature descriptor) to the y_n (n^{th} visual word):

$$u_{mn} = \begin{cases} 1, & n = \operatorname{argmin}||x_m - y_n||_2 \\ 0, & Otherwise \end{cases}$$

4.3.1. Model

The output of encoding is a visual word histogram for each image (Fig. 3). These histograms are used as input features for the learning algorithm and model training. Here, a linear SVM classifier is used for the purpose of classification. The SVM classifier receives as input the visual word histograms, calculated for the training dataset, and draws in the feature space a separating hyperplane between two sets of labeled data by maximizing the margin between the closest points to the hyperplane. These points are known as support vectors. This is equivalent with the following minimization problem:

$$\min \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\varepsilon_i$$

s.t. $y^{(i)}(w^T x^{(i)}+b) \geq 1-\varepsilon_i$, i=1,…..,m ; $\varepsilon_i \geq 0$, i=1,……,m

The parameter C helps balance between two objectives of maximizing margin and reducing misclassification. Higher C values puts more emphasis on achieving lower misclassification rates. The easier approach for solving the optimization problem is the following Lagrangian dual optimization problem:

$$max_\alpha W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}x^{(j)}\rangle$$

s.t. $0 \leq \alpha_i \leq C$, i=1,…….,m

$$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$$

The SVM classifier described above, is used in both of the classification stages that rely on the BOVW technique.

4.4. Texture and Material *Classification*

While the BOVW technique provides a robust and highly accurate means of state classification in this work (section 5), the classification of some of the states with little distinctive feature cannot be accurately achieved using BOVW. For example, a plastered drywall sheet or a painted drywall sheet have surfaces with very few informative local keypoints. As shown in Fig. 2, the texture and material classification is used for two of the four binary classification problems. In one, it is used for visual detection of insulation blankets in images, enabling the categorization of images into *framing* (i.e., no insulation placed) and *insulation* states. The last classification stage also relies on texture classification, where *painted* and *plastered* partitions are separated.

In this work, local binary patterns (LBP) [32] were used as feature for texture classification. To calculate the LBP descriptor, the value of each pixel is compared with a number of its neighbors (Fig. 4c), located at a certain radius around it. These one-to-one comparisons results in a string of binary numbers which represents the observed pattern. Each time the neighbor has a greater value than the center pixel a 1 is recorded, and 0 otherwise. Finally, a histogram is calculated to capture the distribution of patterns over an image (Fig. 4d). Fig. 4 illustrates this process for an image of a plastered wall region.

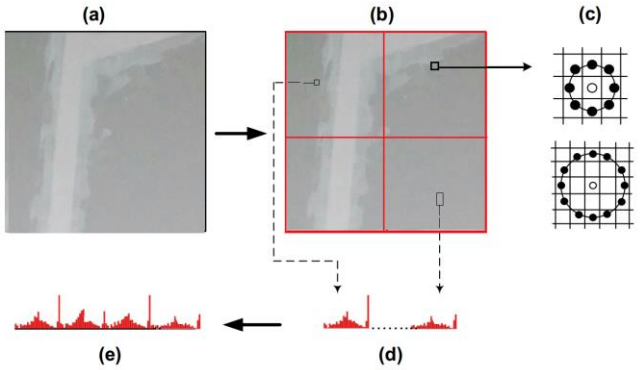

Figure 4. The local binary pattern (LBP) descriptor generation

In the case of insulation blankets, color values were also used as features in addition to the LBP. The pixel values in the *A* channel of LAB color space were used. For both stages of texture classification, the SVM model described in section 4.3.1 was used as the classifier.

5. RESULTS AND DISCUSSION

To evaluate the performance of the proposed technique, three variations of the overall design were considered:

Option 1: BOVW technique for a single step 5-class classification problem
Option 2: cascade scheme (BOVW using SURF features + texture classification)
Option 3: cascade scheme (BOVW using DSURF features + texture classification)

The proposed method was implemented in MATLAB. Existing toolboxes were used for LBP [33], SURF, and DSURF feature extraction [34]. As for the training and model tuning, 60% of the image dataset was randomly selected and used for the purpose of training. Also, 10-fold cross validation was performed. The three variations of the method were tested on the remainder 40% of the images.

The results suggest that the third design option (i.e., BOVW using DSURF features + texture classification) outperforms the other two options. Fig. 5 illustrates the confusion matrix developed based on the tests on testing dataset. For each state of progress (S1 to S5), this matrix indicates the percentages of images classified into five possible states. For example, 95% of images in framing state are correctly classified, while the remainder 5% are misclassified as installed partition.

Please note that S1, S2, S3, S4, and S5 respectively correspond to framing, insulation, installed, plastered, and painted state of progress. Based on the confusion matrix, an average 96% accuracy rate is achieved. The average accuracy rates achieved for all three design options are summarized in Table 2.

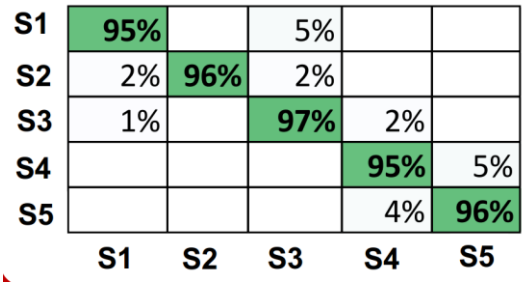

Figure 5. The confusion matrix obtained using the best performing method (third option): S1: framing, …S5: painted partition.

Table 2. The evaluation of various overall design options in terms of average accuracy rate

Method variation	Average Accuracy (%)
Option 1*	73
Option 2**	82
Option 3***	96

*BOVW for 5-class classification
**BOVW using SURF+ texture classification
***BOVW using DSURF+ texture classification

Furthermore, experiments with other texture features such as Gabor filters and histogram of oriented gradients (HOG) [35] was not promising and did not outperform the results achieved using LBP. Also, various variations of the LBP technique were tested and the dominant rotated local binary pattern (DRLBP) [36] resulted in highest accuracy rates (reported in the confusion matrix).

The results (Table 2) were expected due to the fact that last two states of progress (plastered and painted) provide very few informative keypoints for use in BOVW technique. Therefore, it is imperative to take advantage of texture and pattern classification technique. While the DSURF feature extraction significantly increases the run time, this is not an issue in this application domain. Construction firms do not need real-time progress detection, nor it is possible due to the dynamic nature of sites. The method's unsatisfactory performance on design option 2 and a series of bias-validation analysis, encourage the author to explore the use of richer feature vector. Fortunately, the use of DSURF significantly improved the results.

Furthermore, the use of scale and rotation-invariant SURF features enables easier photo capture at dynamic construction sites. This is because images can be captured at varying view points, distances to the wall, and various illumination conditions. It was also observed that the number of local features used for codebook generation has a higher impact on the overall accuracy compared with the length of the codebook. The latter can range between 50 to 500 without jeopardizing the performance.

The proposed solution increases the current accuracy rates for indoor partitions by 20%. It also provides a more robust technique for indoor construction sites, where occlusion and highly cluttered scenes are frequently observed.

8. CONCLUSION

This work introduced a machine learning-based approach toward the detection of state of construction for indoor partitions using digital images. A cascade scheme was introduced which takes advantage of four SVM-based binary classifiers to categorize images into five states of progress. This technique employs a combination of BOVW and texture classification algorithms for more robust detection of progress for indoor partitions. Tests on actual site conditions resulted in over 96% classification accuracy rates.

Future work needs to focus on the improving the current technique in terms of run time, evaluation of other input features to the system, and also design of overall algorithm. It is of interest to the author to extend the application of this work to other stages of construction and increase the number of states classified.

ACKNOWLEDGEMENT

The author would like to extend his gratitude to the CS229 teaching staff for their continued support and hard work. In particular, I would like to thank Nihit, my project TA.

References

[1] Yang, Jun, et al. "Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future." Advanced Engineering Informatics 29.2 (2015): 211-224.

[2] Rezazadeh Azar, Ehsan, and Brenda McCabe. "Automated visual recognition of dump trucks in construction videos." Journal of Computing in Civil Engineering 26.6 (2011): 769-781.

[3] Azar, Ehsan Rezazadeh, and Brenda McCabe. "Part based model and spatial–temporal reasoning to recognize hydraulic excavators in construction images and videos." Automation in construction 24 (2012): 194-202.

[4] Memarzadeh, Milad, Mani Golparvar-Fard, and Juan Carlos Niebles. "Automated 2D detection of construction equipment and workers from site video streams using histograms of oriented gradients and colors." Automation in Construction 32 (2013): 24-37.

[5] Brilakis, Ioannis, Lucio Soibelman, and Yoshihisa Shinagawa. "Material-based construction site image retrieval." Journal of computing in civil engineering 19.4 (2005): 341-355.

[6] Brilakis, Ioannis K., Lucio Soibelman, and Yoshihisa Shinagawa. "Construction site image retrieval based on material cluster recognition." Advanced Engineering Informatics 20.4 (2006): 443-452.

[7] Dimitrov, Andrey, and Mani Golparvar-Fard. "Vision-based material recognition for automated monitoring of construction progress and generating building information modeling from unordered site image collections." Advanced Engineering Informatics 28.1 (2014): 37-49.

[8] Yang, Jun, et al. "Tracking multiple workers on construction sites using video cameras." Advanced Engineering Informatics 24.4 (2010): 428-434.

[9] Park, Man-Woo, and Ioannis Brilakis. "Construction worker detection in video frames for initializing vision trackers." Automation in Construction 28 (2012): 15-25.

[10] Golparvar-Fard, Mani, F. Peña-Mora, and S. Savarese. "D4AR–a 4-dimensional augmented reality model for automating construction progress monitoring data collection, processing and communication." Journal of information technology in construction 14.13 (2009): 129-153.

[11] Kim, Changmin, Hyojoo Son, and Changwan Kim. "Automated construction progress measurement using a 4D building information model and 3D data." Automation in Construction 31 (2013): 75-82.

[12] Kropp, C., Ch Koch, and M. König. "Drywall state detection in image data for automatic indoor progress monitoring." International Conference on Computing in Civil and Building Engineering. 2014.

[13] Kropp, C., et al. "A framework for automated delay prediction of finishing works using video data and BIM-based construction simulation." Proc. of the 14th International Conference on Computing in Civil and Building Engineering. 2012.

[14] Hamledari, Hesam, and Brenda McCabe. "Automated Visual Recognition of Indoor Project-Related Objects: Challenges and Solutions." Construction Research Congress 2016.

[15] Hamledari, Hesam, Brenda McCabe, and Shakiba Davari. "Automated computer vision-based detection of components of under-construction indoor partitions." Automation in Construction 74 (2017): 78-94.

[16] Fei-Fei, Li, and Pietro Perona. "A bayesian hierarchical model for learning natural scene categories." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 2. IEEE, 2005.

[17] Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories." 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Vol. 2. IEEE, 2006.

[18] Yang, Jun, et al. "Evaluating bag-of-visual-words representations in scene classification." Proceedings of the international workshop on Workshop on multimedia information retrieval. ACM, 2007.

[19] Csurka, Gabriella, et al. "Visual categorization with bags of keypoints." Workshop on statistical learning in computer vision, ECCV. Vol. 1. No. 1-22. 2004.

[20] Li, Li-Jia, et al. "Objects as attributes for scene classification." European Conference on Computer Vision. Springer Berlin Heidelberg, 2010.

[21] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer Berlin Heidelberg, 2006.

[22] Lowe, David G. "Distinctive image features from scale-invariant keypoints." International journal of computer vision 60.2 (2004): 91-110.

[23] Bay, Herbert, et al. "Speeded-up robust features (SURF)." Computer vision and image understanding 110.3 (2008): 346-359.

[24] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer Berlin Heidelberg, 2006.

[25] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer Berlin Heidelberg, 2006.

[26] Viola and Jones, "Rapid object detection using a boosted cascade of simple features", Computer Vision and Pattern Recognition, 2001

[27] DENSE SURF: Uijlings, Jasper RR, Arnold WM Smeulders, and Remko JH Scha. "Real-time visual concept classification." IEEE Transactions on Multimedia 12.7 (2010): 665-681.

[28] Wang, Xingxing, LiMin Wang, and Yu Qiao. "A comparative study of encoding, pooling and normalization methods for action recognition." Asian Conference on Computer Vision. Springer Berlin Heidelberg, 2012.

[29] van Gemert, J.C., Geusebroek, J.-M., Veenman, C.J., Smeulders, A.W.M.: Kernel Codebooks for Scene Categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)

[30] Yang, J., Yu, K., Gong, Y., Huang, T.S.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR, pp. 1794–1801 (2009)

[31] Perronnin, F., S´ anchez, J., Mensink, T.: Improving the Fisher Kernel for LargeScale Image Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 143–156. Springer, Heidelberg (2010)

[32] Ojala, Timo, Matti Pietikainen, and Topi Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." IEEE Transactions on pattern analysis and machine intelligence 24.7 (2002): 971-987.

[33] https://www.mathworks.com/help/vision/ref/extractlbpfeatures.html

[34] https://www.mathworks.com/help/vision/ref/detectsurffeatures.html

[35] Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, 2005.

[36] Mehta, Rakesh, and Karen Egiazarian. "Dominant rotated local binary patterns (DRLBP) for texture classification." Pattern Recognition Letters 71 (2016): 16-22.