SEMANTIC SEGMENTATION OF RAILWAY TRACK IMAGES WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Xavier Gibert, Vishal M. Patel, and Rama Chellappa

Center for Automation Research, UMIACS, University of Maryland
College Park, MD 20742-3275, USA
gibert,pvishalm,rama@umiacs.umd.edu

ABSTRACT

The condition of railway tracks needs to be periodically monitored to ensure passenger safety. Cameras mounted on a moving vehicle such as a hi-rail vehicle or geometry inspection car can generate large volumes of high resolution images. Extracting accurate information from those images has been challenging due to the clutter in the railroad environment. In this paper we describe a novel approach to visual track inspection using semantic segmentation with Deep Convolutional Neural Networks. We show that DCNNs trained end-to-end for material classification are more accurate than shallow learning machines with hand-engineered features and are more robust to noise. Our approach results in a material classification accuracy of 93.35% using 10 classes of materials. This allows for the detection of crumbling and chipped tie conditions at detection rates of 86.06% and 92.11%, respectively, at a false positive rate of 10 FP/mile on the 85-mile Northeast Corridor (NEC) 2012-2013 concrete tie dataset.

Index Terms— Semantic Segmentation, Deep Convolutional Neural Networks, Railway Track Inspection, Material Classification.

1. INTRODUCTION

Railway tracks need to be regularly inspected to ensure train safety. Crossties, also known as sleepers, are responsible for supporting the rails and maintaing track geometry within safety ranges. Tracks have been historically built with timber ties, but during the last half century, steel reinforced concrete has been the preferred material for building crossties. Concrete ties have several advantages over wood ties, such being a more uniform product, with better control of tolerances, as well as being well adapted for elastic fasteners, which control longitudinal forces better than conventional ones. Moreover, by being heavier than timber ties, concrete ties promote better track stability [1]. For all these reasons, concrete ties have been widely adopted, specially in high speed corridors.

Although concrete ties have life expectancies of up to 50 years, they may fail prematurely for a variety of reasons, such as the result of alkali-silicone reaction (ASR) [2] or delayed ettringite formation (DEF) [3]. Ties may also develop fatigue cracks due to normal traffic or by being impacted by flying debris or track maintenance machinery. Once small cracks develop, repeated cycles of freezing and thawing will eventually lead to a bigger defects.

This work was supported by the Federal Railroad Administration under contract DTFR53-13-C-00032. The authors thak Amtrak, ENSCO, Inc. and the Federal Railroad Administration for providing the data used in this paper.

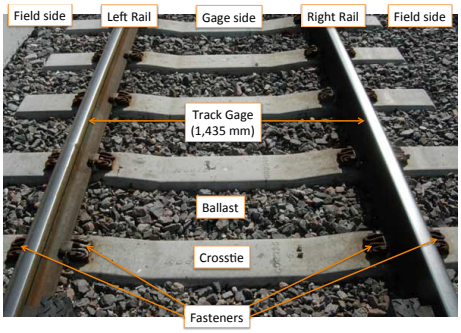


Fig. 1. Definition of basic track elements.

In the United States, regulations enforced by the Federal Railroad Administration (FRA)[1] prescribe visual inspection of high speed rail tracks with a frequency of once or twice per week, depending on track speed. These manual inspections are currently being performed by railroad personnel, either by walking on the tracks or by riding a hi-rail vehicle at very low speeds. However, such inspections are subjective and do not produce an auditable visual record. In addition, railroads usually perform automated track inspections with specialized track geometry measurement vehicles at intervals of 30 days or less between inspections. These automated inspections can directly detect gage widening conditions. However, it is preferable to discover track problems before they develop into gage widening conditions. The locations and names of the basic track elements mentioned in this paper are shown in Figure 1.

Recent advances in CMOS imaging technology, have resulted in commercial-grade line-scan cameras that are capable of capturing images at resolutions of up to 4,096×1 and line rates of up to 140 KHz. At the same time, high-intensity LED-based illuminators available in the market, whose life expectancies are in the range of 50,000 hours, enable virtually maintenance-free operation over several months. Therefore, technology that enables autonomous visual track inspection from an unattended vehicle (such as a passenger train) may become a reality in the not-too-distant future. In our previous work [4, 5] we addressed the problems of detecting and categorizing cracks and defective fasteners. The work described in this paper complements these earlier ones by addressing the problem of parsing the whole track image and identifying its components, as well as finding indications of crumbling and chipping on ties.

This paper is organized as follows. In Section 2, we review some related works on this topic. Details of our approach are given in Section 3. Experimental results on 85 miles of tie images are presented

[1] 49 CFR 213 – Track Safety Standards

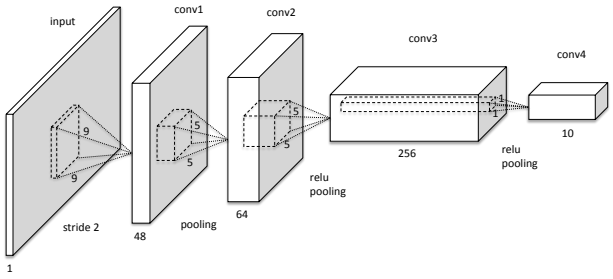


Fig. 2. Network architecture.

in Section 4. Section 5 concludes the paper with a brief summary and discussion.

2. PRIOR WORK

During the last two decades, railways have been adopting machine vision technology to automate the inspection of their track. The very first systems that allowed recording of images of the track for human review and were deployed in the late 1990's[6, 7]. In more recent years, several vision systems have been developed to address different types of inspection needs, such as crack detection [8, 9, 4], defective or missing rail fasteners [10, 11, 12, 5], and missing spikes on tie plates [13]. In [14], Resendiz *et al*. introduced a method for inspecting railway tracks. The system segmented wood ties from ballast using a combination of Gabor filters and a SVM classifier.

The idea of enforcing translation invariance in neural networks via weight sharing goes back to Fukoshima's Neocognitron[15]. Based on this idea, LeCun *et al*. developed the concept into Deep Convolutional Neural Networks (DCNN) and used it for digit recognition[16], and later for more general optical character recognition (OCR)[17]. During the last two years, DCNN have become ubiquitous in achieving state-of-the-art results in image classification[18, 19] and object detection [20]. This resurgence of DCNNs has been facilitated by the availability of efficient GPU implementations. More recently, DCNNs have been used for semantic image segmentation. For example, the work of [21] shows how a DCNN can be converted in to a Fully Convolutional Network (FCN) by replacing fully-connected layers with convolutional ones.

3. PROPOSED APPROACH

In this section, we describe the proposed approach to track inspection using material-based semantic segmentation.

3.1. Architecture

Our implementation is a fully convolutional neural network based on BVLC Caffe[22]. We have a total of 4 convolutional layers between the input and the output layer. The network uses rectified linear units (ReLU) as non-linearity activation functions, overlapping max pooling units of size 3×3 and stride of 2. In our experiments we found that dropout is not necessary. Since no preprocessing at done in the sensor, we first apply global gain normalization on the raw image to reduce the intensity variation across the image. This gain is calculated by smoothing the signal envelope estimated using a median filter. We estimate the signal envelope by low-pass filtering the image with a Gaussian kernel. Although DCNNs are robust to illumination changes, normalizing the image to make the signal dynamic range more uniform improves accuracy and convergence speed. We also

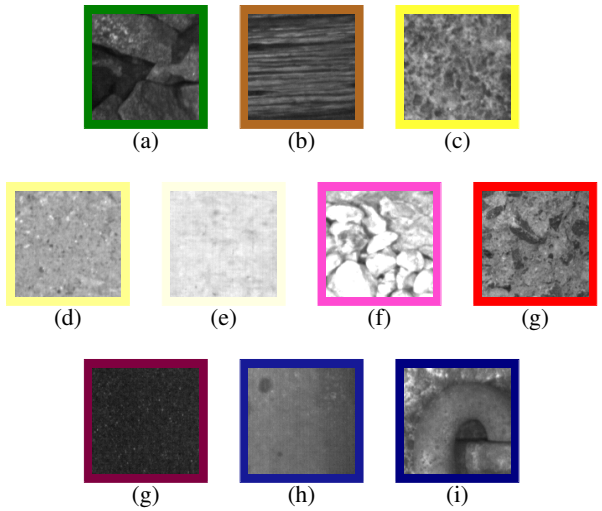


Fig. 3. Material categories. (a) ballast (b) wood (c) rough concrete (d) medium concrete (e) smooth concrete (f) crumbling concrete (g) lubricator (h) rail (i) fastener

Fig. 4. Filters learned at first convolutional layer (range normalized for display).

subtract the mean intensity value calculated on the whole training set.

This preprocessed image is the input to our network. The architecture is illustrated in Figure 2. The first layer takes a globally normalized image and filters it with 48 filters of size 9×9. The second convolutional layer takes the (pooled) output of the first layer and filters it with 64 kernels of size $5 \times 5 \times 48$. The third layer takes the (rectified, pooled) output of the second layer and filters it with 256 kernels of size $5 \times 5 \times 48$. The forth convolutional layer takes the (rectified, pooled) output of the third layer and filters it with 10 kernels of size $1 \times 1 \times 256$.

The output of the network contains 10 score maps at 1/16th of the original resolution. Each value $\Phi_i(x, y)$ in the score map corresponds to the likelihood that pixel location (x, y) contains material of class i. The 10 classes of materials are defined in Figure 3. The network has a total of 493,226 learnable parameters (including weights and biases), of which 0.8% correspond to the first layer, 15.6% to the second layer, 83.1% to the third layer, and correspond to layer and the remaining 0.5% to the output layer.

3.2. Data Annotation

The ground truth data has been annotated using a custom annotation tool that allows assigning a material category to each tie as well as its bounding box. The tool also allows defining polygons enclosing regions containing crumbling, chips or ballast. We used the output of our fastener detection algorithm[5] to extract fastener examples.

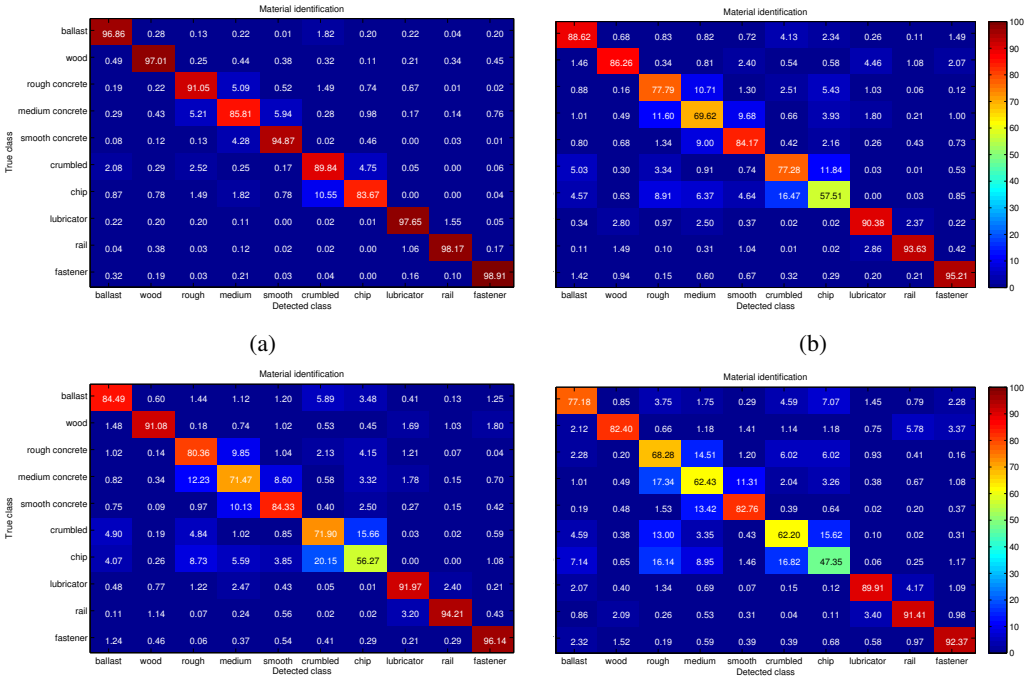


Fig. 5. Confusion matrix of material classification on 2.5 million 80×80 image patches with (a) Deep Convolutional Neural Networks (b) LBP-HF with FLANN (c) $LBP^{u2}_{8,1}$ with FLANN (d) Gabor with FLANN.

3.3. Training

We train the network using stochastic gradient descent on mini-batches of 64 image patches of size 75×75. We do data augmentation by randomly mirroring vertically and/or horizontally the training samples. The patches are cropped randomly among all regions that contain the texture of interest. To promote a robustness against adverse environment conditions, such as rain, grease or mud, we have previously identified images containing such difficult cases and we automatically resampled the data so that at least 50% of the data is sampled from such difficult images.

3.4. Score Calculation

To detect whether an image contains a broken tie, we first calculate the scores at each site as

$$S_b(x,y) = \max_{i \notin \mathcal{B}} \Phi_i(x,y) - \Phi_b(x,y) \quad (1)$$

where $b \in \mathcal{B}$ is a defect class (crumbling or chip). Then we calculate the score for the whole image as

$$S_b = \frac{1}{\beta - \alpha} \int_{\alpha}^{\beta} \widehat{F}^{-1}(t)dt \quad (2)$$

where $\widehat{F}^{-1}$ refers to the t sample quantile calculated from all scores $S_b(x,y)$ in the image. The detector reports an alarm if $S > \tau$, where τ is the detection threshold. We used $\alpha = 0.9$ and $\beta = 1$.

4. EXPERIMENTAL RESULTS

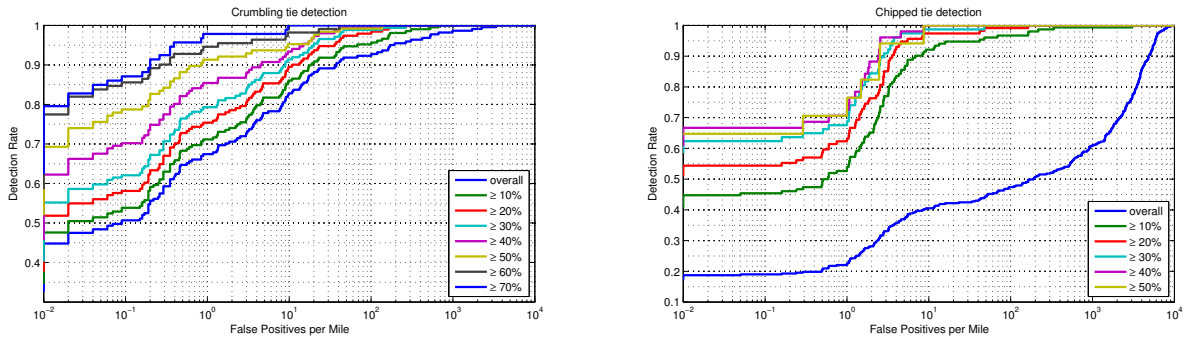
We evaluated this approach on the dataset that we introduced in [5]. This dataset consists of 85 miles of continuous trackbed images collected on the US Northeast Corridor (NEC) by ENSCO Rail's Comprehensive Track Inspection Vehicle (CTIV) between 2012 and 2013. The images were collected using 4 line-scan cameras and were automatically stitched together and and saved into several files, each containing a 1-mile image. As we did in our previous work, we re-sampled the images by a factor of 2, for a pixel size of 0.86 mm. For the experiments reported in this section, we included all the ties in this section of track, including 140 wood ties that were excluded from the experiments reported in [5].

4.1. Material Identification

We divided the dataset into 5 splits and used 80% of the images for training and 20% for testing and we generated a model for each of the 5 possible training sets. For each split of the data, we randomly sampled 50,000 patches of each class. Therefore, for each model was trained with 2 million patches. We trained the network using a batch size of 64 for a total of 300,000 iterations with a momentum of 0.9 and a weight decay of 5×10^{-5}. The learning rate is initially set to 0.01 and it decays by a factor of 0.5 every 30,000 iterations.

In addition to the method described in Section 3, we have evaluated the classification performance using the following methods:

- **LBP-HF with approximate Nearest Neighbor:** The Local Binary Pattern Histogram Fourier descriptor introduced in [23] is invariant to global image rotations while preserving local information. We used the implementation provided by the authors. To perform approximate nearest neighbor we used FLANN[24] with the 'autotune' parameter set to a target precision of 70%.
- **Uniform LBP with approximate Nearest Neighbor** The $LBP^{u2}_{8,1}$ descriptor [25] with FLANN.



(a) (b)

Fig. 6. (a) ROC curve for detecting crumbling tie conditions. (a) ROC curve for detecting chip tie conditions. Each curve is generated considering conditions at or above a certain severity level. Note: False positive rates are estimated assuming an average of 10^4 images per mile. Confusion between chipped and crumbling defects are not counted as false positives.

- **Gabor features with approximate Nearest Neighbor:** We filtered each image with a filter bank of 40 filters (4 scales and 8 orientations) designed using the code from [26]. As was proposed in [27], we compute the mean and standard deviation of the output of each filter and we build a feature descriptor as $f = [\mu_{00}\ \sigma_{00}\ y_{01}\ \ldots\ \mu_{47}\ \sigma_{47}]$. Then, we perform approximate nearest neighbor using FLANN with the same parameters.

The material classification results are summarized in Table 1 and the confusion matrices in Figure 5.

Table 1. Material classification results.

Method	Accuracy
Deep CNN	**93.35**%
LBP-HF with FLANN	82.05%
$LBP^{u2}_{8,1}$ with FLANN	82.49%
Gabor with FLANN	75.63%

4.2. Semantic segmentation

Since we are using a fully convolutional DCNN, we directly transfer the parameters learned using small patches to a network that takes one 4096×320 image as an input, and generates 10 score maps of dimension 252×16 each. The segmentation map is generated by taking the label corresponding to the maximum score. Figure 7 shows several examples of concrete and wood ties, with and without defects and their corresponding segmentation maps.

4.3. Crumbling Tie Detection

The first 3 rows in Figure 7 show examples of a crumbling ties and their corresponding segmentation map. Similarly, rows 4 through 6 show examples of chipped ties. To evaluate the accuracy of the crumbling and chipped tie detector described in Section 3.4 we divide each tie in 4 images and we evaluate the score (2) on each image independently. Due to the large variation in the area affected by crumbling/chip we assigned a severity level to each ground truth defect, and for each severity level we plot the ROC curve of finding a defect when ignoring lower level defects. The severity levels are defined as the ratio of the inspect able area that is labeled as a defect. Figure 6 shows the ROC curves for each type of anomaly. Because of the choice of the fixed $\alpha = 0.9$ in equation (2) the performance is not reliable for defects under 10% severity. For defects that are bigger than the 10% threshold, at a false positive rate of 10 FP/mile the detection rates are 86.06% for crumbling and 92.11% for chips.

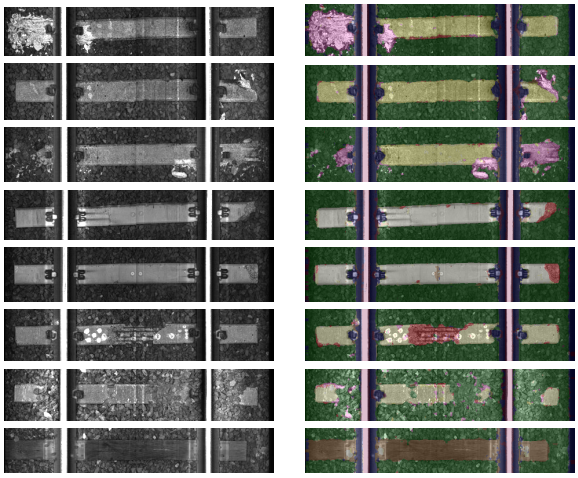


Fig. 7. Semantic segmentation results (images displayed at 1/16 of original resolution)

5. CONCLUSIONS AND FUTURE WORK

Using the proposed fully-convolutional deep CNN architecture we have shown that it is possible to accurately localize and inspect the condition of railway components using grayscale images. We believe that the reason our method performs better than traditional texture features is due to the ability of DCNNs to capture more complex patterns, while reusing patterns learned with increasing levels of abstraction that are shared among all classes. This explains why there is much less overfitting on the anomalous classes (crumbled and chip) despite having a relatively limited amount of training data.

We currently run the network in a feed-forward fashion. In the future, we plan to further explore recursive architectures in order to discover long-range dependencies among image regions with the purpose of better separate normal regions from anomalous ones.

6. REFERENCES

[1] J. A. Smak, "Evolution of Amtrak's concrete crosstie and fastening system program," in *International Concrete Crosstie and Fastening System Symposium*, June 2012.

[2] M. H. Shehata and M. D. Thomas, "The effect of fly ash composition on the expansion of concrete due to alkalisilica reaction," *Cement and Concrete Research*, vol. 30, pp. 1063–1072, 2000.

[3] S. Sahu and N. Thaulow, "Delayed ettringite formation in swedish concrete railroad ties," *Cement and Concrete Research*, vol. 34, pp. 1675–1681, 2004.

[4] X. Gibert, V. M. Patel, D. Labate, and R. Chellappa, "Discrete shearlet transform on GPU with applications in anomaly detection and denoising," *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 64, pp. 1–14, May 2014.

[5] X. Gibert, V. M. Patel, and R. Chellappa, "Robust fastener detection for autonomous visual railway track inspection," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015.

[6] J.J. Cunningham, A.E. Shaw, and M. Trosino, "Automated track inspection vehicle and method," May 2000, US Patent 6,064,428.

[7] M. Trosino, J.J. Cunningham, and A.E. Shaw, "Automated track inspection vehicle and method," March 2002, US Patent 6,356,299.

[8] X. Gibert, A. Berry, C. Diaz, W. Jordan, B. Nejikovsky, and A. Tajaddini, "A machine vision system for automated joint bar inspection from a moving rail vehicle," in *ASME/IEEE Joint Rail Conference & Internal Combustion Engine Spring Technical Conference*, 2007.

[9] A. Berry, B. Nejikovsky, X. Gibert, and A. Tajaddini, "High speed video inspection of joint bars using advanced image collection and processing techniques," in *Proc. of World Congress on Railway Research*, 2008.

[10] F. Marino, A. Distante, P. L. Mazzeo, and E. Stella, "A real-time visual inspection system for railway maintenance: automatic hexagonal-headed bolts detection," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Trans. on*, vol. 37, no. 3, pp. 418–428, 2007.

[11] P. De Ruvo, E. Distante, A.and Stella, and F. Marino, "A GPU-based vision system for real time detection of fastening elements in railway inspection," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 2333–2336.

[12] P. Babenko, *Visual inspection of railroad tracks*, Ph.D. thesis, University of Central Florida, 2009.

[13] Y. Li, H. Trinh, N. Haas, C. Otto, and S. Pankanti, "Rail component detection, optimization, and assessment for automatic rail track inspection," *Intelligent Transportation Systems, IEEE Trans. on*, vol. 15, no. 2, pp. 760–770, April 2014.

[14] E. Resendiz, J.M. Hart, and N. Ahuja, "Automated visual inspection of railroad tracks," *Intelligent Transportation Systems, IEEE Trans. on*, vol. 14, no. 2, pp. 751–760, June 2013.

[15] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 93–202, 1980.

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, November 1998.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Systems (NIPS)*, 2013.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *arXiv:1409.4842*, 2014.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Computer Society Conference on*, 2014.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *arXiv:1411.4038*, 2014.

[22] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv:1408.5093*, 2014.

[23] T. Ahonen, J. Matas, C. He, and M. Pietikäinen, "Rotation invariant image description with local binary pattern histogram fourier features," in *Image Analysis*, pp. 61–70. Springer, 2009.

[24] M. Muja and D.G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *International Conference on Computer Vision Theory and Application VISSAPP'09)*. 2009, pp. 331–340, INSTICC Press.

[25] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[26] M. Haghighat, S. Zonouz, and M. Abdel-Mottaleb, "Identification using encrypted biometrics," in *Computer Analysis of Images and Patterns*. Springer, 2013, pp. 440–448.

[27] B.S. Manjunath and W.Y. Ma, "Texture features for browsing and retrieval of image data," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 8, pp. 837–842, 1996.