Deep Learning Based Image Overlap Detection

Trevor Landeen, *Student Member, IEEE,* and Jacob Gunther, *Member, IEEE*

***Abstract*—Deep learning is attracting a lot of attention because of its success in many research areas. This research is concerned with how deep neural networks may be used to process or compare multiple inputs with an abstract goal. Specifically, this paper addresses the case where the identification of an overlapping region between two images is desired. The scope of this paper includes the conceptual development of different approaches and a discussion of how observing a human's approach to the same task can lead to improved performance. Some results are presented, but a thorough analysis and comparison of the trained DNNs are not included in this paper.**

I. Introduction

DEEP learning has seen a resurgence in popularity in the past decade. Originally introduced in the 1950s, deep learning is a modern revival of artificial neural networks. The interest is being propelled by technological advances breaking down computational and logistical barriers, making the field more accessible than ever before.

The renewed interest is further driven by the impressive results in different application areas. The fields of image classification [1], speech processing [2], and natural language processing [3] have all seen promising results, thus enticing more researchers. In all of these fields, and in many more, deep neural networks (DNN) are being used to interpret the content of the input.

While nontrivial, the problems typically touted as evidence of DNN's superiority over previous approaches are relatively direct; however, there is a lack of research surrounding more abstract problems. In this paper, solutions for a more abstract problem are presented within the scope of an image overlap detection problem. The problem is introduced in more detail in section II. The approaches are presented in section III followed by a performance evaluation in section IV. A discussion of the results and the development of the experiments are in section V.

II. Image Overlap Detection

The image overlap detection problem challenges a DNN to identify which portions of two images are similar. Unlike object detection or classification where the focus is on identifying the content of the images, overlap detection addresses the problem of identifying identical regions within two images.

The difficulty of this problem becomes apparent when the details are considered. When training an image classifier, examples from the same class are always similar. Dogs always look like dogs and cats always look like cats. The same cannot be said for the overlap detection problem. Five example image pairs with complete overlap (for simplicity) are shown in figure 1 with the two images being compared grouped together in columns. The DNN should produce identical outputs identifying the complete overlap for each of the columns; but it is readily visible that the five pairs are not visually similar. There are different textures, colors, patterns, and structures in each; thus, this problem is significantly more abstract than object detection.

A. Related Work

The body of research most closely related to the image overlap detection is often referred to as patch matching. Many of the patch matching networks are structured after the siamese neural network introduced in 1994 by Bromley et al. in a signature verification problem [4]. The basic siamese neural network is one containing two processing branches (usually identical, but not required) to process two inputs simultaneously.

The patch matching problem essentially asks the neural network with deciding if two input images are fundamentally the same. MatchNet [5] uses a siamese neural network to learn image representations necessary to determine if two images are the same. Melekhov et al. approach the matching problem in a similar manner but use Euclidean distance between two learned representations to make a decision [6]. Simo-Serra et al. also used the Euclidean distance metric for similarity but intended to replace SIFT features in other algorithms with the features learned by the siamese network. Bailer et al. approached patch matching for optical flow using a siamese structure using the a relaxed Euclidean distance cost function [7]. Other uses of siamese neural networks in patch match related applications are given in [8][9][10].

III. Proposed Approach

Information concerning the end goal or any prior beliefs must be communicated to the DNN somehow. In this paper, the target output format, the output format encodings, and the cost functions are discussed. Three different models are presented in section III-C.

A. Overlap Mask

A binary mask is used to represent the overlap between the two input images. Let the two images be referred to as the reference image and the test image. The overlap mask is used to indicate the overlap with respect to the reference image. Figure 2 shows the relationship between the reference image, test image, and overlap mask. In the overlap mask, the white region (represented with a '1') is the region of overlap and the black region (represented with a '0') is the region of *no* overlap.

Representing the overlap using a mask naturally quantizes the potential overlaps and the dimensionality of the mask is significant. The mask used in this research is a 10×10 matrix

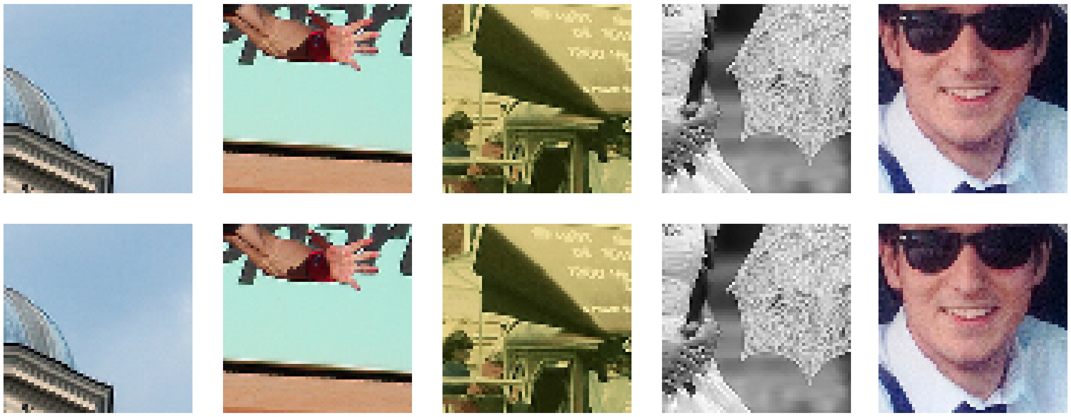


Fig. 1. Example challenge with overlap detection. For simplicity, the images are identical (100% overlap). The columns are image pairs and the rows are different samples. The output of the DNNs should be identical for each of the five image pairs, even though the content of each pair varies greatly. All images are cropped from source images in the COCO dataset.

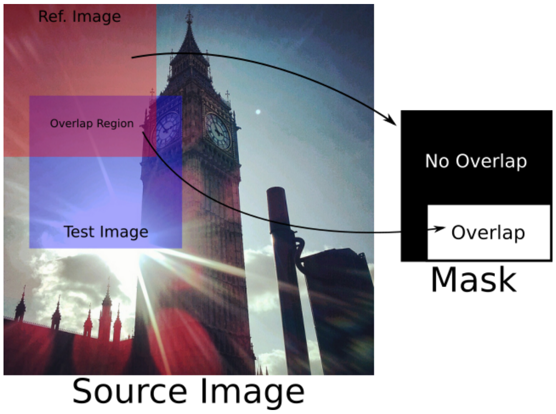


Fig. 2. Demonstration of the overlap mask. The reference image (red) and the test image (blue) are taken from the same source image. The overlap mask is defined with respect to the reference image. In the overlap mask, white regions indicate overlap and black indicates no overlap. Source image from the COCO dataset.

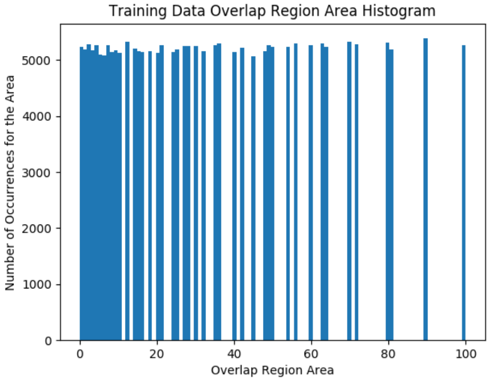


Fig. 3. Histogram of the training dataset overlap sizes. The samples in the dataset are approximately uniform over possible overlap sizes for a 10×10 mask.

yielding 56 possible overlap sizes with 362 possible, unique, overlap representations. Determination of these numbers is simply a counting problem and not within the scope of this paper.

B. Datasets

Deep neural networks require separate training and testing datasets. There are no preexisting, publicly available datasets having two input images and the overlap mask. As a result, both datasets were generated by sampling reference and test images from source images obtained from the COCO dataset [11].

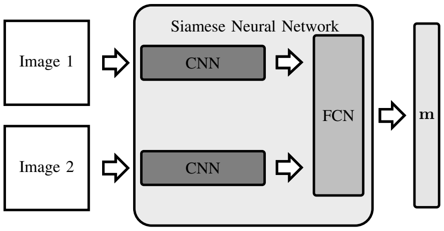
Both datasets were generated to contain an approximately uniform distribution of the 56 possible overlap sizes. A histogram of the training dataset overlap sizes is given in figure 3.

The training dataset consists of 224,400 reference and test image pairs; the testing dataset consists of 28,050 image pairs.

C. Three Encodings

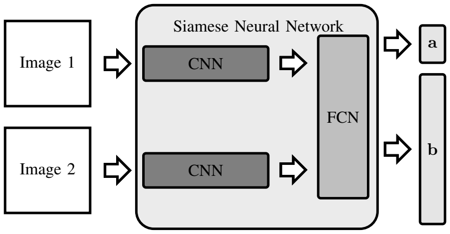
In this section, three different encodings of the overlap mask matrix are presented. The DNN processing prior to the output encoding is similar for all three encodings. Two images are input into a siamese neural network followed by a fully connected network (FCN). The overlap mask encoding is only present on the output of the FCN. Each of the three networks are trained separately and as such, all have learned different weights.

1) Direct Encoding: The first encoding presented is a simple, straightforward, representation of the overlap mask



$$M = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{10}] \in \{0,1\}^{10\times 10}$$
$$\text{vect}(M) = \mathbf{m} = [\mathbf{c}_1^T, \mathbf{c}_2^T, \ldots, \mathbf{c}_{10}^T]^T \in \{0,1\}^{100\times 1}$$

Fig. 4. Overview of the direct encoding used. The DNN outputs a single vector representing every element in the overlap mask. The columns of the overlap mask are stacked to form a single column vector representing the entire mask.



$$\mathbf{a} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \qquad \mathbf{b} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \qquad \overset{M}{\begin{bmatrix} a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_k & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ a_2 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & a_3 \end{bmatrix}}$$

Fig. 5. Overview of the corner encoding network used. The DNN outputs two one-hot vectors, each representing one of the corners of the overlapping regions. For example, if $a_3 = 1$ and $b_k = 1$, then the overlapping region is the rectangular region bounded by those two corners.

and is shown in figure 4. The overlap mask is represented as a matrix M with columns $\mathbf{c}_i$. The vector representation of the overlap mask, $\mathbf{m}$, is found by stacking the columns of M into a single vector.

2) Corner Encoding: The second encoding considered in this research is intended to provide rectangular shaped, predicted overlap regions. Instead of predicting the mask directly, the DNN uses two output vectors representing two corners of the overlapping region. After the corner predictions are made, computing the rectangular region bounded by the corners is trivial. The representation of this encoding is presented in figure 5. The output vector $\mathbf{a}$ represents which of the four main corners is in the overlapping region and the output vector $\mathbf{b}$ indicates where in the mask the interior corner of the overlap occurs. Both $\mathbf{a}$ and $\mathbf{b}$ are one-hot vectors, meaning in each vector, every element is zero with a single exception equal to one.

3) Row-Column Encoding: The third encoding developed in this research is intended to provide a rectangular shaped, predicted overlap region like the corner encoding but provide a much simpler model. The row-column encoding is shown in figure 6. Two output vectors, $\mathbf{r}$ and $\mathbf{c}$, are used to represent the rows and columns where overlap occurs. An element of $\mathbf{r}$ equal to one indicates that overlap occurs somewhere in that row. Similarly, an element of $\mathbf{c}$ equal to one indicates that overlap occurs somewhere in that column. The complete overlap is simply the intersection of the nonzero elements of $\mathbf{r}$ and $\mathbf{c}$.

IV. Results

DNNs using the three overlap mask encodings were all trained with Keras [12] using the Tensorflow [13] backend. Early-stopping was implemented to limit over fitting during training. Training was performed using the Adam optimizer [14].

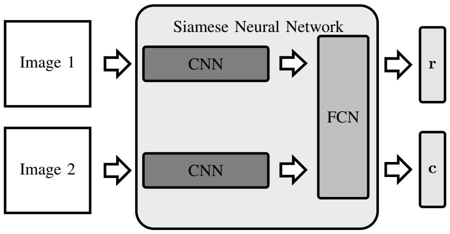
Different loss functions were used during the training of every DNN so the prediction accuracy is used as a comparison metric. The overlap mask is a 10×10 matrix so the prediction accuracy is simply the number of elements in the mask correctly predicted as either one or zero. The prediction accuracy is averaged over all 28,050 image pairs to provide a single scalar value for comparison.

A. Direct Encoding

The DNN using the direct encoding achieved an average prediction accuracy of 79.6%. A single sample from the testing dataset is presented in figure 7. This single sample is not necessarily representative of the performance over the entire dataset; however, the shape of the predicted overlap region is (the white region). The true mask is rectangular but the predicted mask is not. It is in the correct general region but it does not have the rectangular shape. Furthermore, the nonoverlapping region is not contiguous. These observations led to the development of the corner encoding.

B. Corner Encoding

The DNN using the corner encoding achieved an average prediction accuracy of 79.61%. As before, a single sample from the testing dataset is presented in figure 8. The performance for the sample shown is not representative of all samples but it does demonstrate a fundamental flaw of the corner encoding. Recall that the output of the network is two vectors which represent the corners of the overlap region. In this example, the interior corner was almost predicted correctly but the main corner was predicted incorrectly. As a result, the predicted overlapping region and the true overlapping region do not have any areas in common. Because of this flaw, the row-column encoding was developed.



$$\mathbf{r}\ \begin{bmatrix}0\\0\\1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\ \overset{M}{\begin{bmatrix}0&0&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&0&0&0\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1&1\end{bmatrix}}$$

$$\underset{\mathbf{c}}{\begin{bmatrix}0&1&1&1&1&1&1&1&1&1\end{bmatrix}}$$

Fig. 6. Overview of the row-column encoding network used. The DNN outputs two vectors, each representing the overlap along the rows or the columns. If an element in **r** is 1, then the overlap occurs somewhere in that row. The overlap mask is formed by taking the intersection of the rows and columns.

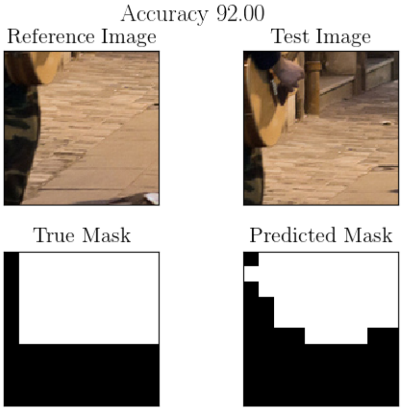


Fig. 7. Example output from the DNN using the direct overlap mask encoding for one of the testing samples. The upper left image is the reference image. The upper right image is the test image. The lower left is the true overlap mask (with respect to the reference image), and the lower right is the predicted overlap mask. This example shows the inability of DNNs using this encoding to predict rectangular shaped overlap regions.

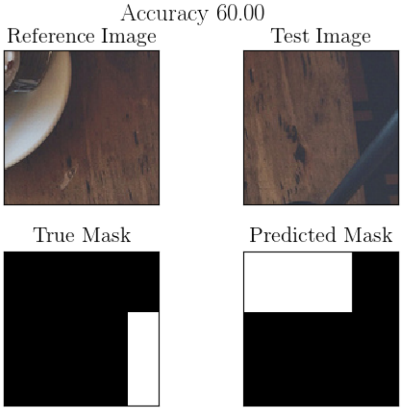


Fig. 8. Example output from the DNN using the corner overlap mask encoding for one of the testing samples. The upper left image is the reference image. The upper right image is the test image. The lower left is the true overlap mask (with respect to the reference image), and the lower right is the predicted overlap mask. This example demonstrates a flaw with the corner encoding approach.

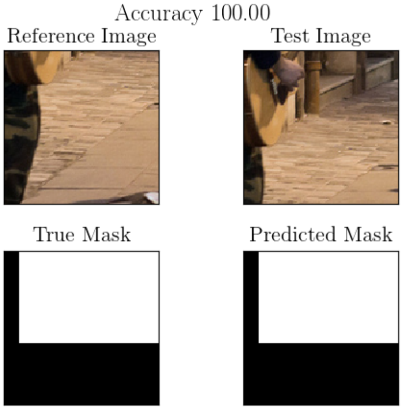


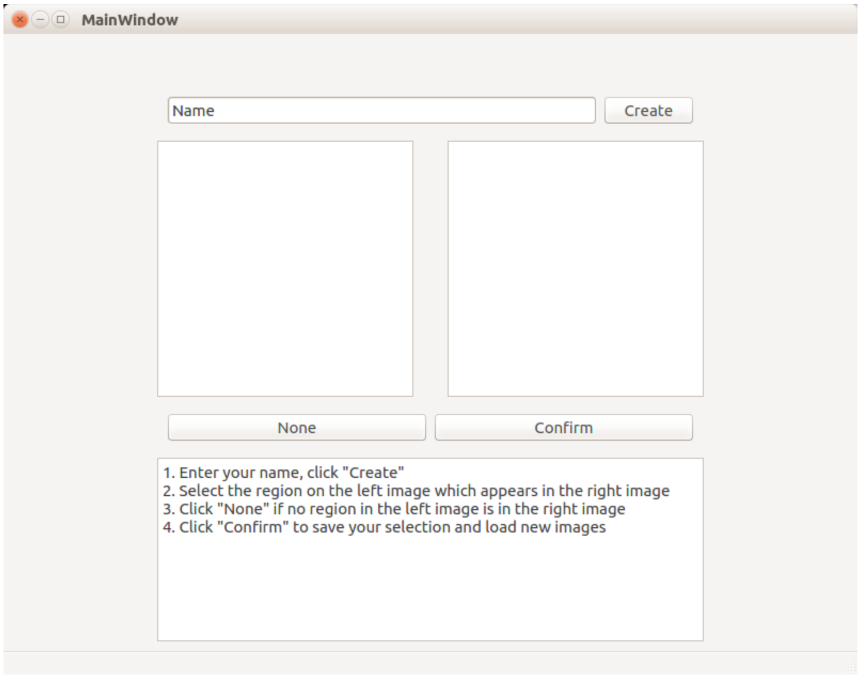
Fig. 9. Example output from the DNN using the row-column overlap mask encoding for one of the testing samples. The upper left image is the reference image. The upper right image is the test image. The lower left is the true overlap mask (with respect to the reference image), and the lower right is the predicted overlap mask. This example demonstrates the ability of DNNs using this encoding to provide rectangular overlapping regions.

C. *Row-Column Encoding*

The DNN using the row-column encoding achieved an average prediction accuracy of 84.84%. The single sample from the testing dataset is shown in figure 9. In this instance, the DNN perfectly predicted the overlapping region; however, this does not occur for every sample. This DNN is able to make better overlap mask predictions on average than any of the other networks or encodings tested. The overlapping and nonoverlapping regions were contiguous in every sample tested and furthermore, the overlapping regions were always rectangular.

V. Discussion

The three encodings presented in this paper were not developed simultaneously. Initial attempts were limited to the direct overlap mask encoding and were focused on network architecture and cost functions. During the evaluation of the trained DNNs (most of which are not included in this paper), the question of how well a human could perform the same



1. Enter your name, click "Create"
2. Select the region on the left image which appears in the right image
3. Click "None" if no region in the left image is in the right image
4. Click "Confirm" to save your selection and load new images

Fig. 10. GUI used to let humans solve the same image overlap detection problem. Two images are displayed and the user selects the portion in the left image also present in the right image.

task arose. In seeking to quantify the human performance, a simple software program was developed in an effort to make observations. The user interface for the program is presented in figure 10.

The functionality of the program is simple. After the user enters a name, a reference image and a test image are displayed. Using the mouse, the user selects the portion of the reference image found in the test image. The software will give each user a few tests and once completed, the results. At this point in time, the sample size is not large enough to draw any conclusions; instead, this software was able to provide insight into how a human would approach the same task.

During the development and testing of the software, it became apparent that the middle of the overlap region was not necessary to correctly identify the overlapping area. Identifying the boundary of the overlap was sufficient, but there were not any implementations of the direct encoding doing this. Additionally, it became apparent that humans had an upper hand because the software forced them to select rectangular regions. Due to these observations, other encodings were investigated.

The development of the corner encoding models followed the direct encoding models. During the trained model evaluation process, the issue with selecting the wrong corner was apparent and motivated the investigation which led to the row-column encoding.

Only three trained DNNs were presented in this paper and 12 other trained DNNs are out of the scope of this paper and not included. Overall, nearly every DNN using the row-column encoding outperformed the DNNs using the direct encoding. The nearly five percentage point improvement seen with DNNs presented in this paper is enough to demonstrate the clear advantage the row-column encoding has over the others.

VI. Conclusion

The image overlap detection problem has been approached before from an analytical standpoint, but not with deep neural networks. The patch match problem is similar but is limited to single binary decisions for the input images as a whole. The object classification networks which are helping to drive the rise in popularity of DNNs lack the abstraction that the image overlap detection problem possesses.

The results briefly summarized in this paper demonstrate the potential of DNNs to solve abstract and less defined problems. The three different output encodings demonstrate that the way the training data is presented to the DNN is related to how well the DNN is able to learn the given task. In this case, a different encoding with a much simpler loss function was able to outperform a more complicated loss function with a simpler encoding.

Perhaps one fo the most significant observations is the role the developed software program played. Developing and testing the software provided insights which led the solutions resembling the human-like approach. In developing a deep learning solution, there may be valuable knowledge to be gained from examining the human approach.

Future work in this research is two pronged. The collection of human performance data and its analysis will provide a quantified performance benchmark for model comparison. The other branch of research focuses on dealing with more complicated image orientations. In the present stage of research, the images are purely shifted and contain no rotations, perspective shifts, or illuminance variations; future research will address these cases.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[2] A. Graves, A. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *CoRR*, vol. abs/1303.5778, 2013. [Online]. Available: http://arxiv.org/abs/1303.5778

[3] Y. Goldberg, "A primer on neural network models for natural language processing," *CoRR*, vol. abs/1510.00726, 2015. [Online]. Available: http://arxiv.org/abs/1510.00726

[4] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Advances in Neural Information Processing Systems 6*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. Morgan-Kaufmann, 1994, pp. 737–744. [Online]. Available: http://papers.nips.cc/paper/769-signature-verification-using-a-siamese-time-delay-neural-network.pdf

[5] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "Matchnet: Unifying feature and metric learning for patch-based matching," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3279–3286.

[6] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching," in *2016 23rd International Conference on Pattern Recognition (ICPR)*, Dec 2016, pp. 378–383.

[7] C. Bailer, K. Varanasi, and D. Stricker, "Cnn-based patch matching for optical flow with thresholded hinge embedding loss," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2710–2719.

[8] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 4353–4361.

[9] J. Zbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," *CoRR*, vol. abs/1510.05970, 2015. [Online]. Available: http://arxiv.org/abs/1510.05970

[10] D. Chung, K. Tahboub, and E. J. Delp, "A two stream siamese convolutional neural network for person re-identification," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 1992–2000.

[11] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[12] F. Chollet *et al.*, "Keras," https://github.com/keras-team/keras, 2015.

[13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980