Image similarity with Deep Neural Networks

Sagit Efrat Dan Semiat
{sagite, dannysem}@campus.technion.ac.il
Technion — Israel Institute of Technology

July 12, 2017

Abstract

Image Similarity is a well investigated problem over the last decades. There are many ways to encounter the problem (Rui, Huang, & Chang, 1999), that is a joint work of database management and computer vision communities. Recent works take use of *Deep Learning* methods, that has been proven to handle image representation very well. *Deep Neural Networks* are used for variety of tasks including image recognition, natural language text and audio processing, that showed a significant progress with this approach, comparing to other techniques.
In this review we shall focus on resemblance of images based on their feature vectors, using *Deep Learning* methods for the feature extraction task and discuss the challenges.
The outline is as follow: First we focus on a specific method with a specific dataset (Wang et al., 2014), Then we take the acknowledgements from previous step and test it on the targeted data we received from the client.

1 Scientific and Academic background

Search-by-example, i.e. finding images that are similar to a query image, is an indispensable function for modern image search engines. In order to compare similarity of two images, one can calculate the distance between the two - this depends on the chosen distance metric. When dealing with heavy images there is a need to reduce dimensionality, so an image can be represented by a smaller set of features - a feature vector. In the past, most solution to the image similarity problem had use hand crafted features. In recent years *Machine Learning* methods are used to improve performance (LeCun et al., 1989). Current *Deep Neural Networks* achieve remarkable performance at a number of vision tasks surpassing techniques based on hand-crafted features (Donahue et al., 2014; Krizhevsky, Sutskever, & Hinton, 2012; Zhou, Lapedriza, Xiao, Torralba, & Oliva, 2014; Donahue et al., 2014). For a similar set of problems such as Classification the state-of-the art are *Deep Neural Networks*. Classification belongs to a sub-field of *ML* called supervised learning where Image retrieval belongs to the sub-field of unsupervised learning, hence the use in state-of-the-art techniques differ and need some adjustments.

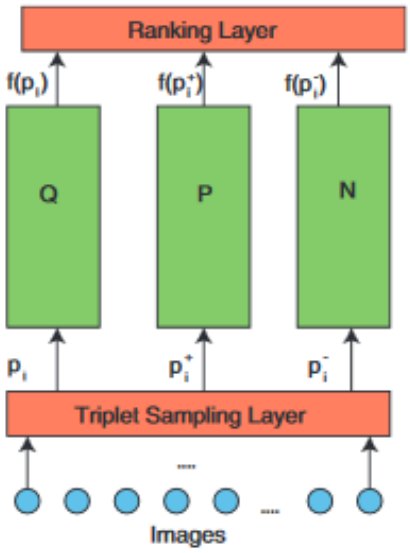


Figure 1: Deep Ranking model

2 First Step - Proof of Concept

2.1 Deep Ranking Method

In their article (Wang et al., 2014), present a new model for the task of image similarity. They created a dataset where each column is a triplet of images: query, positive, and negative (we use the notation q, p+, p- respectively) and a joint label of the images in each triplet, total of 15K different images. The positive image is the more similar to the query than the negative image, according to a human ranker [1]. Their network takes triplets of image as input, every object of the triplet is fed independently into three identical deep neural networks with shared architecture and parameters. A ranking layer on the top evaluates the hinge loss of a triplet. The ranking layer does not have any parameter. During learning, it evaluates the models violation of the ranking order, and back-propagates the gradients to the lower layers so that the lower layers can adjust their parameters to minimize the ranking loss. (Figure 1) The inspiration of this project came from this article, we wanted to use a dataset that is more similar to our problem - noisy unlabelled images, and use the tools we learned for the famous datasets: *IMAGENET*, *CIFAR10*, *MNIST*.

2.2 Our Tests

As first stage we set the baseline - we experiment with different deep neural networks architectures, and compare it with the article results. This will determine which architecture and distance metric combine give fair results on the 'toy' dataset:

- *VGG 19* - with feature vector in the size of 4096.
- *VGG 16* - with feature vector in the size of 4096.

[1] The data is available at *https : //sites.google.com/site/imagesimilaritydata*

- *VGG_CNN_M_2048* - with feature vector in the size of 2048.
- *VGG_CNN_M_1024* - feature vector in the size of 1024.

All models were pre-trained on *CIFAR10* dataset, since it was shown that a trained neural network on this dataset learned diverse features. We neglected the thought of training our net on the client's dataset since we wanted to avoid over-fitting. After exploring the clients data, each is in different form - construction sites, field, parking lots, etc. and they might need to retrieve different kind of objects: cars, trees, ladders, safety vests, hats, goggles, tractors, cherry picker, etc.
The measurement metrics we examined[2]: *sqeuclidean*, *correlation*, *hamming*, *cosine*, *matching*, *dice*, *canberra*, etc.
For each image in the dataset we extracted its feature vector accordingly for each examined architecture. Then, for a query image we ranked all images by their similarity where it was determined by the nearest vector according to a measurement method.

The goal is to ensure that the positive image ranked more similar than the negative image. In the Table 1 (below) is a heat-map graph where each point represents a triplet: the ranks of the positive image - x and the negative image - y to each query image. A low rank means more similar and a high rank means less similar to the query.

The desired graph should be more dense above the x=y line, and sparse below. After exploring the results, we can see that the best characteristic for the task are *VGG 16* model, and the *sqeuclidean* and *hamming* distance measurement, that together showed better results on the triplets data.

2.3 Evaluation

In order to evaluate our algorithm success we wanted to compare the results with the results of (Wang et al., 2014). Their method of Similarity precision is as follows: each triplet is a member of a specific class. In order to rank similarity of the positive and negative images to a query, they sample 1000 images from the class of the query. Then the samples are sorted and each get a rank that symbolizes how similar it is to the query. If the rank of p+ is higher than the rank of p- then its considered a success.

Our further research handles unlabelled data, hence, there is a need to remove the dependency of the class labels, therefore we ranked all of the images and not just the ones that are in the same class. As a result we expect less accuracy. The actual results are around 60% success, it doesn't appear very promising, so we explored the 'failures': In Figure 2 are shown the most extreme comparisons, where in region A we expect to see very good similarity, in regions B, D we expect to see no relative connection between the query and the p+, p- and in region C we expect to see examples of when a human eye can be mistaken in labelling the similarity. All the examples are visualized in Table 2.

[2]Explanations can be found here: *https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance.pdist.html*

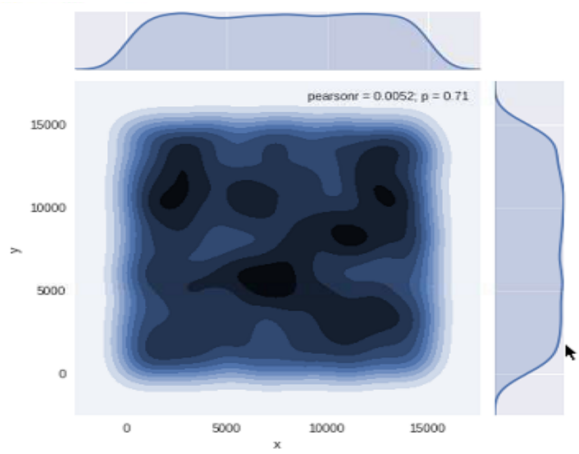
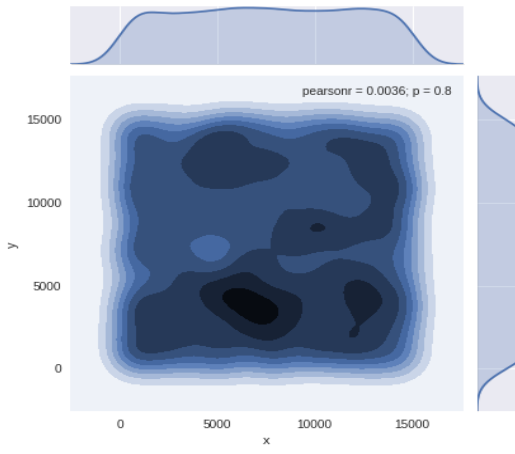
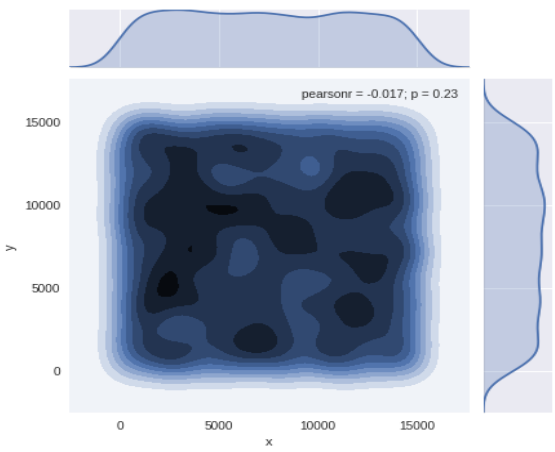
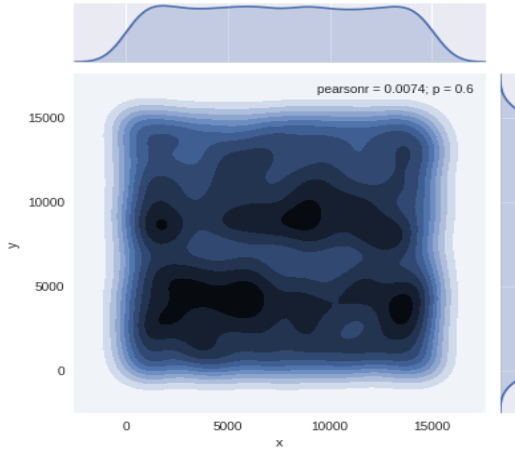
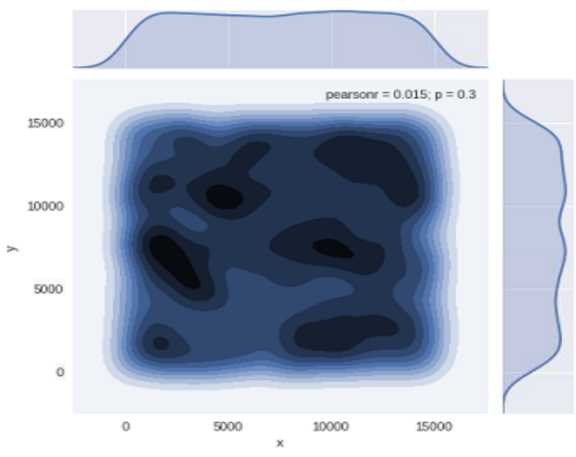
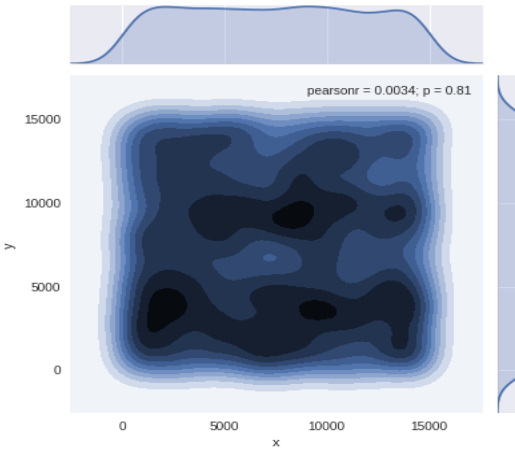
Architecture	Sqeuclidean	Hamming
VGG 16		
VGG 19		
VGG_CNN 1024		

Table 1: Comparing different Neural Networks architectures with 2 distance metrics: Square Euclidean, Hamming

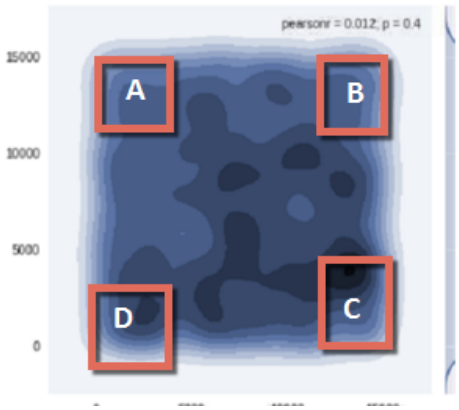


Figure 2: extreme comparisons

3 Second Step - experiments on the client's data

3.1 Dataset

The dataset consists of variety of image quality and categories, mainly industrial related, approx. 10K large images. From those we extracted crops of different object, the size of each crop is independent, approx. 17K count. Sum of 27K images.

In the previous step we used a pre-trained model and got good results, we expect to take the knowledge to this step, hoping to see satisfying results on the client's data.

3.2 Our Tests

The experiments are now done using the same *VGG 16* model, and we examined two distance measurements: *sqeuclidean* and *hamming*. We performed step 2 in order to demonstrate the model behaviour in the company's data

3.3 Evaluation

In order to evaluate our algorithm success we wanted to compare the query image to the top ranked result images. In Table 4 are the results. We can see that *sqeuclidean* measurement gives a much better success rate than the *hamming* on client's data, which means it's the best measurement function we've discovered to handle the application.

We measured the results with a human observer - telling if the resulting images are similar to the query image by containing the same object, similar shapes, colours, etc. *sqeuclidean* gave a correctness rate of 86% success.
hamming gave a correctness rate of 56% success.

The VGG 16 model gave a much better success rate on the dataset using *sqeuclidean* rather than *hamming*. We should only try the better model on the next step.

4 Third Step - Decision making on the client's data

4.1 Dataset

50 hand picked cropped images that serves as query images, approx. 10K large images, each in different size (10K images in total). As opposed to former step, we now distinguished between the two image groups - in order to find the closest similar image from this dataset out of the real dataset (without crops).
For each query image, we ranked all images according to the similarities.

4.2 Evaluation

In order to evaluate our algorithm success we wanted to compare the query cropped image to the top ranked result images. The VGG 16 model gives a correctness rate of 86% success on the test data. Comparing to previous steps, in this step we expected worse results. That is because we used data consisted of only large images, which may be distorted when inserted into the model, when comparing to the input query cropped image. Having said that, it is clearly a success that we received this correctness rate.

5 Discussion and Next Step

We performed step 3 in order to demonstrate the model behaviour in 'real' data search, as opposed to the much more detailed dataset we had before - which had different image sizes and cropping. Overall, while considering the failures, the model seems to identify various objects using the feature extraction application. Taking into account the unique basic and complex shapes, colours, textures, angles, etc. Some of the incorrect similarities may be a result of the cropped image's surrounding, which may have affected the general decision in the image's feature vector. Another reason is that the texture in a part of the cropped image may be extremely unique, which resulted in 'out-of-the-box' image similarities.

This method is relevant to other tasks like finding changes in streaming information such as videos (Long et al., 2016). Many companies use video footage for the creation of labeled images, this method can tell the labeller (human or computer) when the stream frames differ and a new label is required.

References

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning* (pp. 647–655).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541–551.

Long, G., Kneip, L., Alvarez, J. M., Li, H., Zhang, X., & Yu, Q. (2016). Learning image matching by simply watching video. In *European conference on computer vision* (pp. 434–450).

Rui, Y., Huang, T. S., & Chang, S.-F. (1999). Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation*, *10*(1), 39–62.

Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., ... Wu, Y. (2014). Learning fine-grained image similarity with deep ranking. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 1386–1393).

Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in neural information processing systems* (pp. 487–495).

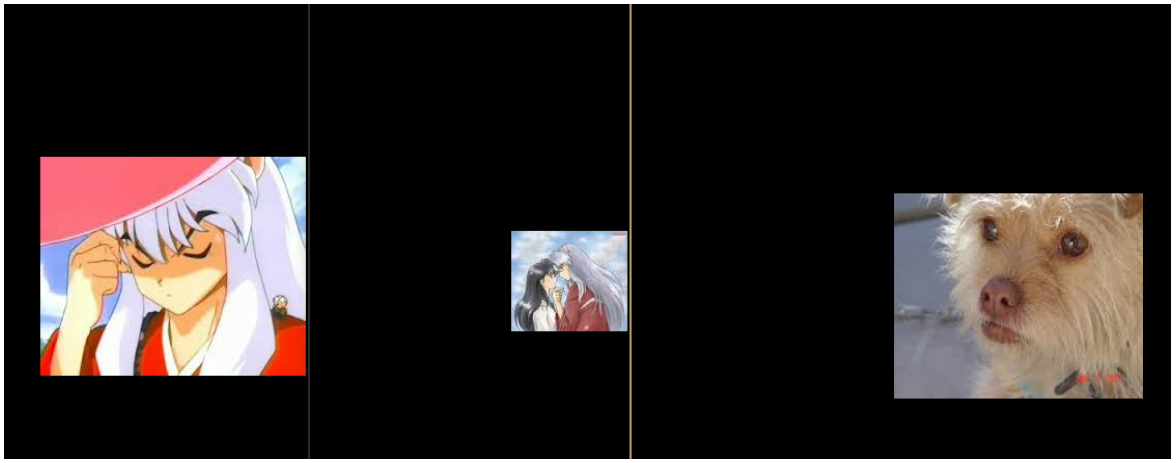
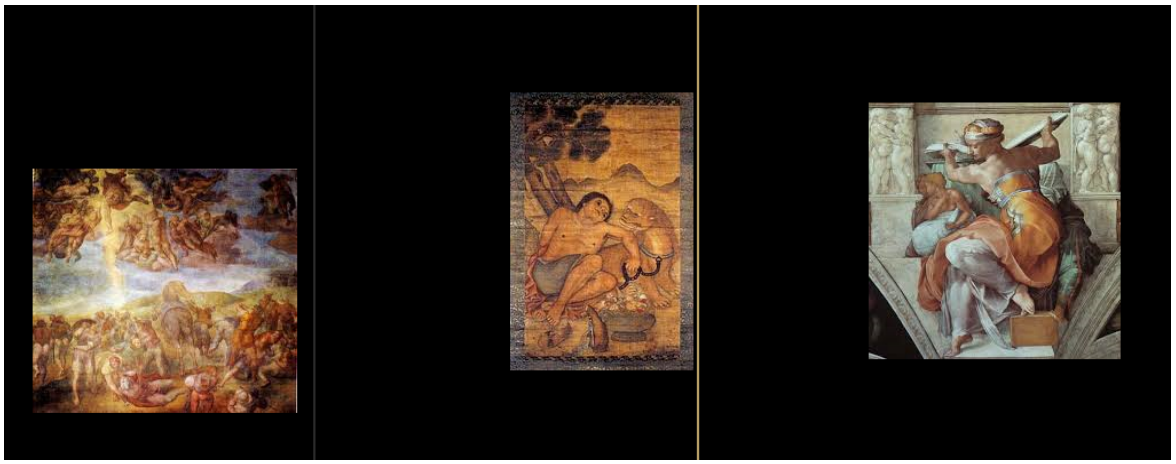
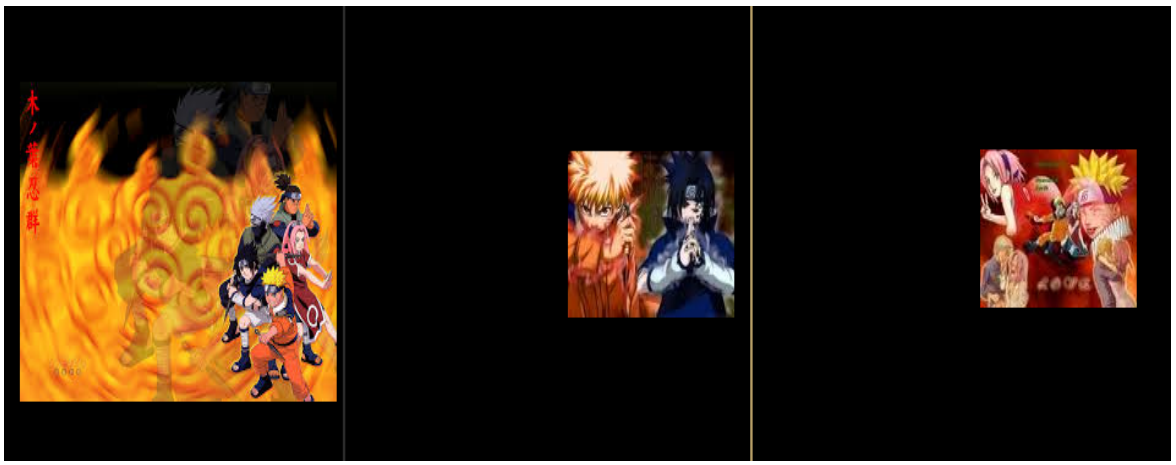
Box	Query,Positive,Negative
A	
B	
C	
D	

Table 2: Examining the extreme comparisons according to heat-map regions

Box	Sqeuclidean measurement method
good results	
bad results	

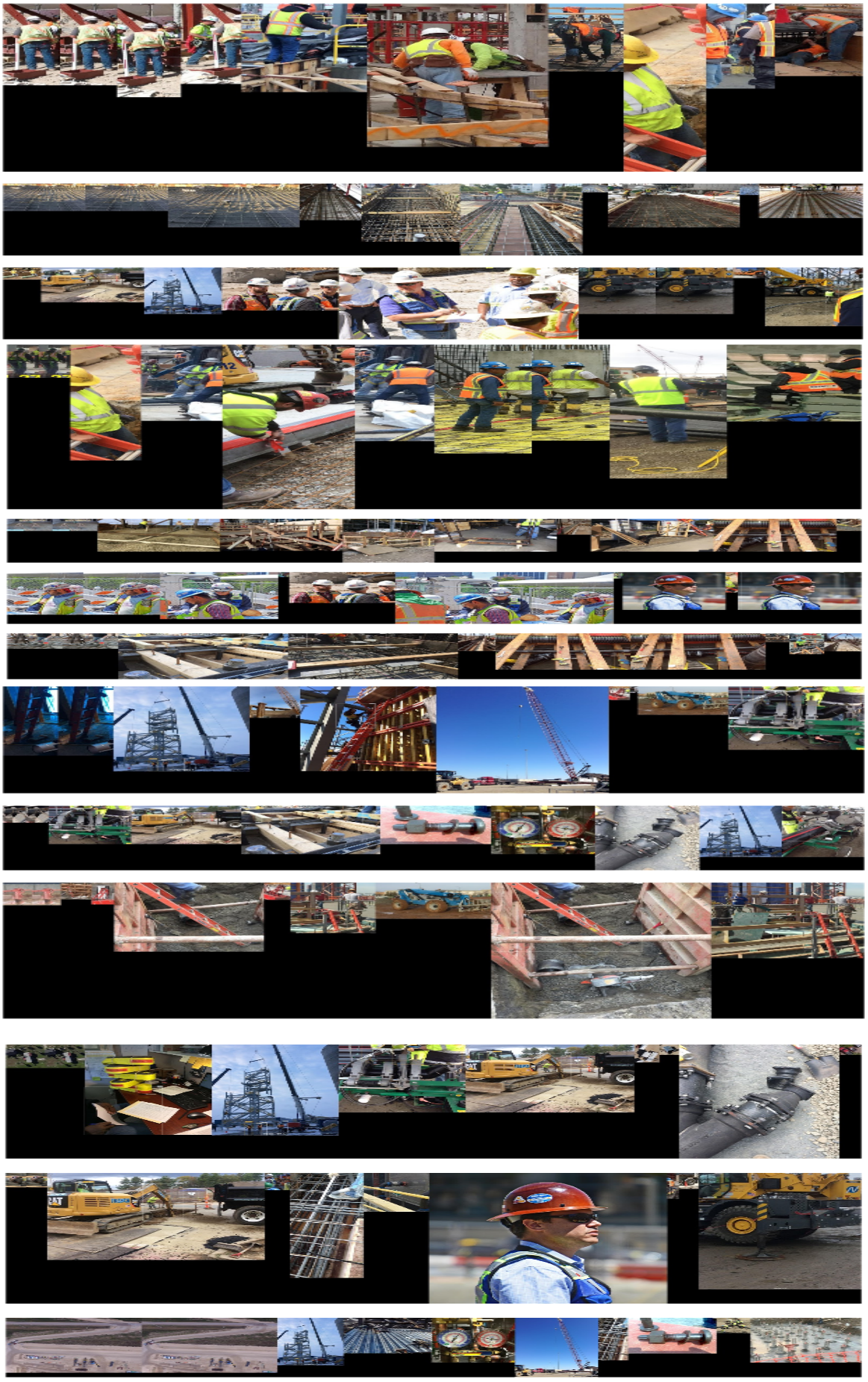
results	Hamming measurement method
good	
bad	

Table 4: Exploring VGG 16 with *sqeuclidean*and *hamming*distance metrics for the 'real' data

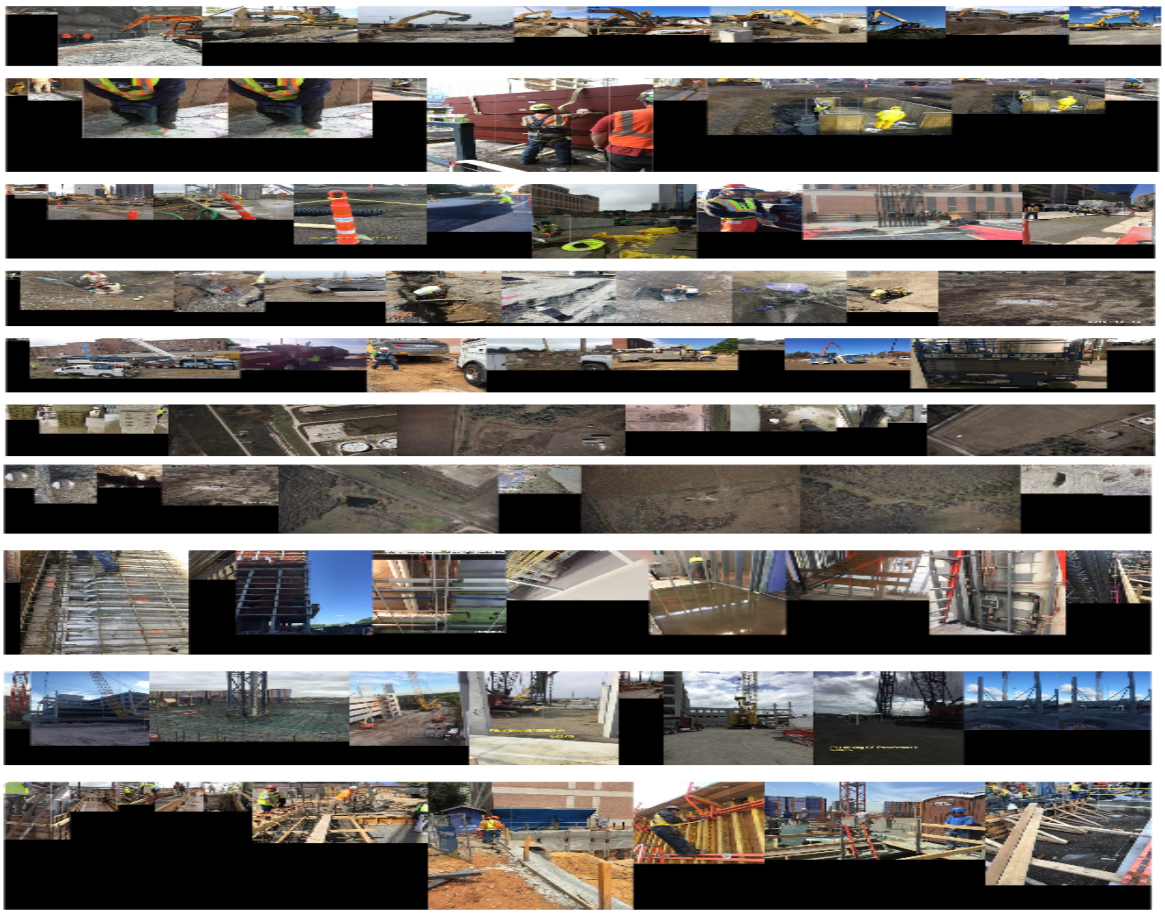
Box	Sqeuclidean measurement method
good results	
bad results	

Table 5: VGG 16 and sqeuclidean for the hand picked crops