Using CNNs for Low Level Vision

Raghavender Sahdev

Department of Electrical Engineering and Computer Science and Centre for Vision Research
York University
Toronto, Canada
sahdev@cse.yorku.ca

***Abstract*—In this paper we present a literature survey of the usage of Convolutional Neural Networks (CNNs) for generating local image patch descriptor, comparing image patches and its application to depth computation. The survey reviews current works for the same. The idea behind the survey is to show that CNNs can not only be used for high level classification and detection tasks but also used for low level vision tasks like computing local image descriptors which can act as a better representation of the image patches than the traditional hand crafted descriptors like SIFT, HOG, SURF, etc. We present some preliminary work for descriptors generations and then present its application like Image Retrieval and Depth estimation.**

Keywords-Convnets; disparity; CNNs; descriptors; image retreival;

I. INTRODUCTION

Convolutional Neural Networks have been used lately as a standard for getting high accuracies on tasks like Object Detection [11], Image Classification [12], Feature Learning, Segmentation [13], Tracking in videos [14]**,** etc. In this paper we present a literature review on Convolutional Networks being used in context of a low level vision task - image patch matching and its application to disparity/depth estimation and image retrieval. Low level vision tasks like finding a robust descriptor is a fundamental task in Computer Vision. If one can find good solutions to this problem other high level vision tasks can be tackled in more efficient manner. So far the descriptors have been limited to traditional approaches like SIFT [15], HOG [16], SURF [17], etc. In this literature survey we present some of the work indicating the power of CNNs to learn descriptors from a large dataset.

First we give a brief over view of approaches being currently proposed for comparing image patches and descriptor generation using convnets, then go on to discuss the estimation of depth using Convnets and present some work which does image ranking and image retrieval. Common thing among both of these tasks is the matching of small image patches which minimize a cost function and compute the similarity between images. Some approaches learn the similarity metric while others use standard metrics like the Euclidean norm.

We divide the paper into five sections. Section II gives an overview of computing descriptor and local patch matching approaches. Section III presents some of the work related to image retrieval which has tasks like computing patch descriptors and matching inherent in it. Section IV presents approaches for computing depth from Stereo Images and monocular images. Finally we provide a conclusion of this review paper in Section V and provide possible future lines of work.

II. PATCH MATCHING USING CONVNETS

Learning image features from local image patches is ubiquitous in most of the computer vision applications and forms the basis for applications like stereo, structure from motion, wide baseline matching, pose estimation, detection, classification, recognition and many others. Traditional handcrafted approaches SIFT [15], HOG [16], SURF [17] have been outperformed by CNN based models in most of the applications listed above.

Early works of using CNNs for finding local key point descriptors was discussed in [1]. Jahrer et al. [1] present a convent model to perform two tasks- a classification task and an image patch matching task using a Siamese network. They use a convent model of 8 layers with alternating convolutional and subsampling layers as used by LeCun in [18]. For the classification task they have an output of 600 units which classifies an image patch into a particular key point as shown in figure 1 (i). Additionally they also propose a Siamese network (figure 1 (ii)) which they use for matching a pair of image patches. The network computes a feature descriptor representation for each of the input images and computes the matching similarity score based on a cost function.

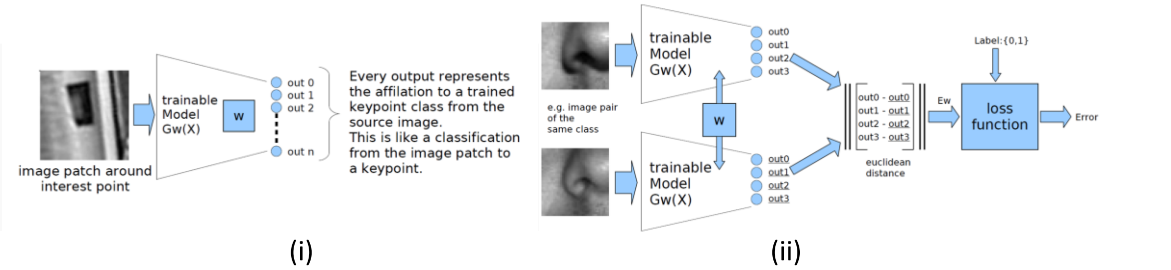
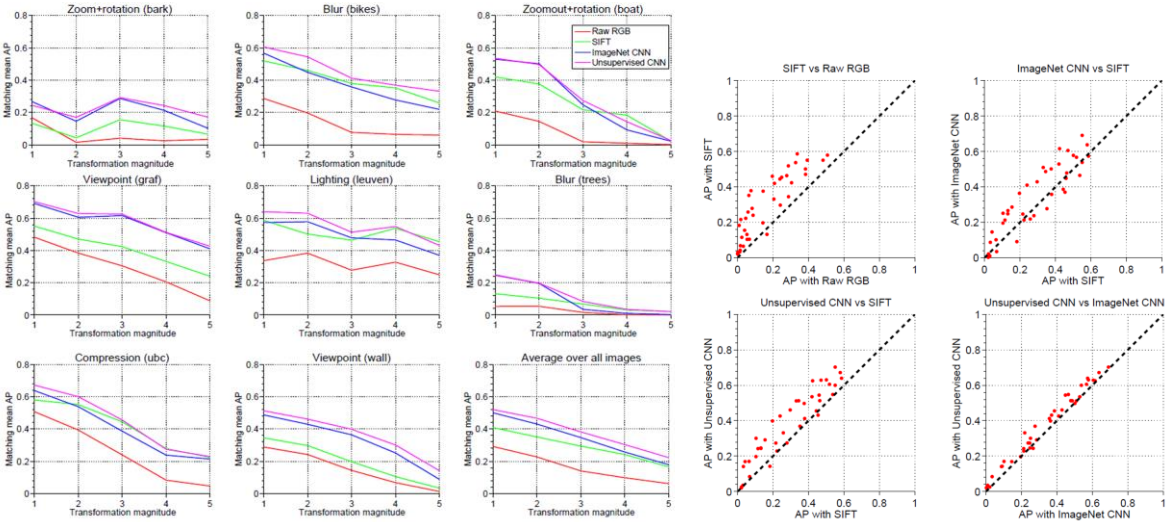


image patch around interest point
trainable Model Gw(X)
w
out 0
out 1
out 2
out n
Every output represents the affilation to a trained keypoint class from the source image.
This is like a classification from the image patch to a keypoint.
e.g. image pair of the same class
trainable Model Gw(X)
w
trainable Model Gw(X)
out0
out1
out2
out3
out0
out1
out2
out3
out0 - out0
out1 - out1
out2 - out2
out3 - out3
euclidean distance
Ew
Label:{0,1}
loss function
Error
(i)
(ii)

Figure 1. i). Trainable model used for patch classification and ii). the siamese network used in [1].

Similar to Lowe's SIFT descriptor [15], Jahrer et al. [1] also use DoG (Difference of Gaussian) for keypoint detection. For detecting stable keypoints, they generate random perspective wraps and knowing the homography, they detect similar keypoints in these wraps. This also takes care of invariance to orientation. They introduce some jitter in position, scale, orientation and grayscale values to get a better training set for the classifier. For the classificaiton task, they match keypoints based on the closest match of the output of the convnet to the source image's keypoints. For the image patch matching they use the closest euclidean distance to get a match. Finally they evaluate their proposed model on three datasets and get better results than the state of the art descriptor – SIFT. They collected training data with approx. 600 keypoints in the source image, every keypoint class had at least 150 images of 60x60 size in 4 scales. This gave over 100,000 training examples for training their model. They show results in the form of ROC curves and show that SIFT is outperformed by the CNN descriptor. This work presented a naïve use of a convnet without much modifications to the CNN architecture which were done later by Krizhevsky in 2012 [12].

A similar work was presented by Fischer et al. [2] where they too showed that features obtained from Convnets outperform SIFT in the low level task of desriptor matching. They used supervised and unsupervised CNN models for image matching and showed that the unsupervised model slightly outperformed the supervised one. To build the Supervised model they used an ImageNet pre-trained model of AlexNet [12]. This model included Max-pooling, local response normalization, Rectified Linear Units (ReLUs) and dropout which the Convnet model used by [1] did not have. For doing unsupevised learning they used a smaller network than AlexNet to avoid overfitting; they used 3 convolutional layers and 1 fully connected layer with the first two convolutional layers being followed by max-pooling. They used ReLUs and dropout but did not use local response normalization. For training this unsupervised model they used random images from Flickr and extracted 16,000 seed patches of size 64x64. To increase this dataset they applied 150 elementary random transformations to each of the seed patches. Transformations included rotation, scale variation, color, contrast variation and blur. In this paper they test if the features obtained from a convnet trained on a classification task could perform as well as a local interest point descriptor. For emperical evaluation they compared their approach to the state of the art local descriptor – SIFT and used raw RGB image patch to act as a naïve baseline. They used the matching dataset by Mikolajczyk et al. [19] which contains 48 images. To increase the dataset they used an additional 416 images obtained from 16 Flickr images by subjecting each image to transformations. They used this for training the supervised model. For measuring the performance they extracted elliptic regions of interest and corresponding image patches from both images using the maximally stable extremal regions (MSER) detector [20]. Then they extracted descriptors from patches and used the minimum euclidean distance to match them. Average precision plots of 4 different approaches – SIFT, Raw RGB, supervised CNN and the unsupervised CNN were plotted. It was observed that the average precision for the unsupervised Convnet outperformed all others. Interestingly the unsupervised CNN improved performance over SIFT as much as SIFT improves performance over raw RGB patches. Figure 2 shows this result. Both Jahrer et al. (2008) [1] and Fischer et al. (2014) [2] showed that surprisingly convnets can be also used for low level vision tasks like descriptor matching. Jahrer's network [1] from 2008 lacked some optimizations (ones later proposed by Kritchvezsky [12]) which were present in Fischer's work. Additionally Fischer's model had higher computational efficiency than Jahrer's work. Also it was observed that in both [1] and [2] SIFT was computationally more efficient than the CNN based approach. However SIFT could not perform as well as the convnet. SIFT took 2.95 ms and unsupervised CNN took 37.6 ms.



(i) (ii)

Figure 2. Mean Average Precision and Scatter Plots for different pairs of descriptors on the Mikolajczyk[19] dataset. Each point in the right image corresponds to an image pair.

Long et al. [7] questioned the fact if Convolutional neural networks trained for classification and detection tasks with larger receptive fields could be actually considered as a viable choice for low level vision tasks like Alignment, Key point classification / prediction. The aim of this work was similar to that of [2]. Long et al. [7] use a CNN model similar to AlexNet [12] with the order of response normalization and pooling layers reversed.The model was pre-trained on ImageNet for the classification task. For visualizing features, they perform non-parameteric reconstruction of images from features. The image patches in an image are replaced by the average of the top-k nearest neghbors in a convnet feature space. Features are matched using a cosine similarity measure. Their database contains 1 million patches from PASCAL VOC 2011. Fig 1 and 2 from [7] can be seen for a visualization of this. It has not put here due to space constraints. They also perform the task of Intraclass alignment based on *conv4* features. They use an MRF smoothness prior to allow all images to be warped into alignment. They compare their results for alignment taks with SIFT flow [21] and NN transfer and get better results than them. Keypoint classification is performed to check given an image and coordinates of a keypoint on that image can a classifier be trained to label the keypoint. For this task they extract features at each keypoint using SIFT and using the column of each convnet layer whose receptive field center lies closest to that keypoint. A one vs all linear SVM was trained usig SIFT. It is seen in this task too convnet perform better than SIFT highest performance comes from conv4 and conv5 layer. Finally they perform keypoint prediction task where again the convnet outperforms SIFT. It should be noted that in each of the mentioned tasks the convet was trained for the classification tasks and still outperformed the best handcrafted feature – SIFT. It is expected that the performance in each of the above tasks would increase if the CNN model is separately trained for it. The conclusion from Long et al. [7] is similar to that from [1] an [2], CNNs can be trusted for low level vision tasks too and that they outperform SIFT significantly.

Zagoruyko and Komodakis [3] proposed different CNN models to compute similarity between image patches. Similar to the siamese model used in [1], they proposed different convnet architectures based on the siamese models. They proposed the siamese model and the pseudo-siamese model. In a siamese model, there are two branches with the same set of weights of the convolutional kernels as in [1]. These weights are learnt from the input image patches to come up with a descriptor for each of the two imput patches. Then these descriptors from each branch are fused to the top layer which consists of fully connected and ReLU units. This top layer acts as a similarity function to compute the similarity between the two input patches. The pseudo-siamese model differs from the siamese in terms of the weights in each of the branches being separately learnt. Additionally in [3] the authors also propose a 2 channel network in which the input patches are given to the same convnet and a similarity score is computed. This network is faster to train than the siamese based networks. They also proposed a siamese based central surround two stream architecture which receives input as the resized input patch and the central patch extracted from the each of the input images. So finally four inputs are fed to two siamese based models incorporated in the central surround two stream architecute; in the top layers the descriptors generated from each of the four networks are fused to compute a similarity score for the initially given two image pairs. Another interesting architecture proposed is the Spatial pyramid pooling network for comparing image patches. In this model a spatial pyramid pooling layer is attached as the terminal layer before the final descriptor for each patch is computed to ensure the same dimensionality of the feature computed before it is fed to the similarity computation network. This network has the advantage over the other proposed networks in the sense that it does not restrict the given input images to have the same size. This kind of behaviour was not observed in the CNN models proposed in [1],[2] or [4]. Data augmentation was carried out on the training dataset to improve leaning. For each proposed CNN model in [3] three types of evaluations were performed – i) local image patches simlarity on the dataset by [3] ii) wide baseline stereo matching iii)local descriptor performance evaluation. Initially they do an empirical evaluation of their proposed CNN models on the standard bench mark dataset [22]. This dataset contains three subsets (Yosemite, Notredam and Liberty) each of which contains 450,000 image patches. They plot ROC curves and report the average errors obtained in each of the proposed CNN models. It was seen that the 2 channel based networks were the top performing ones. Other proposed models performed better than the handcrafted state of the art (SIFT) technique for descriptor matching. For wide baseline stereo matching also they achieved better than state of the art results. For doing local descriptors performance evaluation they use the same dataset used by [2] which is same as Mikolajczyk dataset [19]**.** The approach presented by [3] is built upon the basic approach used by the siamese based network of [1]. The contribution of [3] is the variability in the architecture design of different CNN models. Additionally they present an extensive experimental evaluaion of their approach to challenging datasets and applications which was missing from the work by [1] and [2].

Han et al. [4] built a convnet architecture which they call MatchNet to do image patch matching. MatchNet was designed for a general wide baseline viewpoint invariant matching which is different from the local matching problem in stereo. MatchNet is inpired from AlexNet [12] and has 2 networks built – Feature Network and Metric Network. Feature Network computes the feature reperesentation of a local image patch and Metric Network which consists of 3 fully connected layers with a softmax layer computes the similarity between the two images. MatchNet has the two towers of feature networks with same weight parameteres which compute the descriptors from the image patches and these features are then fused by the metric network. MatchNet is similar to the siamese based model proposed by [1] and [3] with some architectural differences in terms of the number of convolutional and pooling layers. MatchNet does not use Local Response Normalization and dropout but uses ReLU as the non linearity for the convolutional layers. At the end of the feature network towers is attached a bottleneck layer to reduce the dimensionality of the feature representtation and prevent overfitting. For training the network, MatchNet minimizes a cross entropy error. The training dataset is sampled in way to ensure that the negative non matching pairs and the positive matching pairs are equally balanced to avoid biasing the network to the negative class. For computing features, the feature tower/network and the metric network are used separately. First all image patches are run throught the feature tower and then the image pairs are fed into the metric tower to compute the similarity between the image pairs. This is done to avoid running the same image patch through the feature tower multiple times. For experimental analysis they tested their model the same dataset used by Zagoruyko and Komadakis [3] which is the UBC patch dataset [23]. They compare their results with the SIFT base line. For computing similarity from a pair of SIFT features they use the L2 linear SVM on 128 dimensional element wise squared difference features and two layer fully connected neural networks in the 256d SIFT concatenation. For MatchNet they evaluate the similarity score by varying the dimensions of the 2 fully connectred layers in the metric network. It was seen that MatchNet significantly outperformed SIFT baseline and also the previous state of the art method proposed by Simonyan et al. [24]. The 2channel based model proposed by Zagoruyko and Komadakis [3] achieved slightly better results on an average on the UBC (Notredame, Yosemite and the Liberty) dataset. MatchNet computes compact feature representations than the 2channel based model and has an additional bottleneck layer to vary the dimensionality of the featuer representation. In this sense both models proposed of [3] and [4] have their pros and cons. [3] presents different CNN model architecture and achieve better results than MatchNet model. On the other hand MatchNets feature representation is variable and compact than the 2channel based CNN model of [3]. Both approaches used in [3] and [4] outperform the Simonyan's et al. [24] work.

Simo-Serra et al. [5] propose a siamese based architecture which uses 2 CNNs with shared parameters to compute feature point descriptors from each of the input local patches. Their proposed model generalizes well towards applications for which the model was not trained. The desctiptors produced are robust to rotation, scaling, varying illumination, view point changes and non-rigid deformation. Contrary to the approach used by MatchNet[4] and [3] which learn a similarity metric, the approach presented in [5] does not rely on a trained metric for computing similarity and instead uses a simple L2 norm which is a standard in comparing local image patches. This ensures that the desctriptors generated from their model is not task specific as in [3] and [4]. Their CNN model contains 3 convolutional layers with each layer having a filter layer, non linearity, pooling and a normalization layer. Hyperbolic tangent (tanh) is used and L2 pooling for the pooling layers. They tried various architectures and found this architecture produce optimal results. No reasoning has been presented in the specific choice of such a model. Normalization is important for descriptors and hence has been used. For training the input patches are sampled for positive and negative classes initially. On the sampled patches, hard positive and negatives are used; these are the ones which incur a high loss function; this is done to increase the discriminative capability of the learnt desctriptors. For training they used the Multi view correspondence dataset [22]. It is the same dataset used in [3] and [4]. The proposed architecture considerably outperforms handcrafted SIFT and the previous state of the art method used by VGG based model of [24]**.** For a detailed evaluation the reader is encouraged to look at Figure 4,5 of [5] and Table 4 in [5]. It has not been included here due to space constraints. The approach presented in [5] mainly differs from other approaches in terms of the dimensionality of the learnt descriptors which is much more compact (128D) than those produced in [3], [4] and [25]. They used an intelligent approach of using hard negative to increase the discriminative power of the model. Most papers used for computing local image descirptors use a network based on the siames architecture originally proposed by Bromley et al. [26]. The basic approach used in each of the paper is using a siamese based networks to compute local descriptor representaiton of the image patch and then using a traditional similarity metric like euclidean norm [1], [2] or the L2 norm [5] or a learnt similarity metric specific to a task [3],[4]. Each of the architecure differs in terms of the filter sizes, number of convolutional layers, non linearity used (ReLU vs tanh), pooling layers, etc. Approaches used do not give exact reasons of using a particular model other than trying various models and using the one giving best performance.The different CNN models used in [3] do give a slight indication of when to use one over the other like using the Spatial pyramid pooling based conv net for variable input patches comparison.

An interesting work is that of Dong and Soatto [34] who show in their paper a modification to the SIFT descriptor which they call DSP-SIFT (domain size pooling SIFT). They modified SIFT based on pooling gradient orientations across different domain sizes in addition to spatial locations. It should be noted that to come up with the DSP-SIFT descriptor no training is required and the dimensionality of the feature vector is 128 the same as SIFT which is smaller than what CNN based model generate. They showed that DSP-SIFT outperformed convolutional neural network of Fischer et al. [2] by 28.29%. Detailed results of their proposed descriptors can be seen from Figure 3 and 4 from their paper [34]. They also claim that Convnet descriptor have an advantage of having high dimensionality and hence more representative power, yet their DSP-SIFT of 128 dimensions outperforms [2] by a large margin. They did something similar to what CNN did to the traditional SIFT by getting higher accuracies than SIFT; DSP-SIFT got better accuracies than PhillipNet of [2]. This is something indicating convnet may not be the ideal choice for a low level vision task. One of the encouring papers which combats CNN based approaches being applied to low level vision as not too representative which is opposing to the idea behind this literature survey of using CNNs for low level vision.

III. IMAGE RETRIEVAL

Image retrieval systems in general have the 3 steps i) interest point detection ii) interest point description and iii) local patch matching. The goal of step (i) is to find points which are reproducible under different scale and view-points. Description of those interest points refers to finding a robust feature representation around those keypoints. Various CNN based approaches to finding and computing interest point description were discussed in the previous section. Approaches for computing local patch similarity between images were discussed in [1], [3], [4] and [7] in the previous section. Here we describe some convnet based approaches which use the 3 steps to build an efficient image retrieval system. Paulin et al. [47] propose a Convolutional Kernel Network (CKN) to compute local features from an image patch and perform unsupervised training to address the process of doing image retrieval. The contribution of Paulin et al [47] is that of generating a family of patch descriptors Patch-CKN based on Convolutional Kernel Networks and showing that for the application of patch and image retrieval it is possible to learn competitive patch level descriptors without supervision. The concept of a convolutional kernel network was initially introduced by Mairal et al. [41]. For the 3 steps of image retrieval pipeline, they use a Hessian Affine Detector [42], for computing the point descriptor, the local patch's feature representation in Euclidean space is computed and finally for patch matching they encode patch descriptors and aggregate them into a fixed length image descriptor using the VLAD representation [43]. They use convolutional features to encode fixed size image patches (size 51x51) using unsupervised learning through CKNs. In this work they use a convet model same as was used in the Phillip Net discussed in [2] which had 3 convolutional and one fully connected layers. A convolutional kernel network has the same architecture as that of a CNN the difference being the filters in a CNN are learnt and defined in a data dependent manner where as a CKN computes a feature representation based on a kernel feature map and learns the parameters of the map in a data independent manner. A patch descriptor is computed using subsampling and stochastic gradient optimization. They define a single layer kernel definition and a multi-layer CKN kernel. A multi-layer CKN representation can be thought of having a kernel laid on top of another to encourage learning of better feature representation. They define 3 types of CKNs – CKN raw, CKN white and CKN grad. CKN white preprocesses each sub-patch of the CKN's first layer by subtracting their mean color and uses PCA whitening whereas the CKN grad is fully invariant to color. The CKN raw directly receives input from the raw RGB patch to the network. They use the Mikolajczyk [44] dataset and generate another dataset which they call the Rome Patches dataset. On both the datasets they use a Hessian Affine Detector for extracting interest regions. They use a VLAD representation to aggregate the features obtained from the local patches to compute a global feature representation of the image. For validating their proposed CKN model they use 2 CNNs – AlexNet [12] and PhillipNet [2] with the same parameter settings as used in the original CNN model. They then use CKNs to learn the filter maps and train their model in an unsupervised manner. Finally they report accuracies obtained by training their model. They validate their approach against 4 datasets Holidays, UKbench, Oxford dataset [45] and the Rome Patches dataset. They report the results of their approach and compare it with other state of the art approaches based on CNNs and VLAD+SIFT. Through the results shown in the table 3, 4, 5 of [51], one can see that their CKN based trained network outperforms the current CNNs. In this paper they proposed a new descriptor Patch-CKN for patch and image retrieval and showed that their descriptors perform at par on better than the supervised CNNs on standard patch and image retrieval benchmark datasets.

Wang et al. [6] built a CNN model to compute fine grained image similarity to rank the image for the process of image retrieval. They use a set of triplets (query image, positive image and a negative image) as an input for their model. The image similarity relationship is characterized by relative ordering in the triplets. It should be noted here that a CNN model trained for image classification may not work well for an image retrieval task. They also propose a novel multiscale network structure which has a convnet with two resolution paths. One of these paths learns the visual appearance of the image. Their goal is to compute the similarity between two images P and Q according to their squared euclidean distance in the embedding space:

$$D\big(f(P), f(Q)\big) = ||f(P) - f(Q)||_2^2$$

Where $f(.)$ is the image embedding function that maps an image to a point in an Euclidean space, and $D(.,.)$ is the squared euclidean distance. The objective is to minimize this distance for a similar pair of images. For computing this distance they employ a pairwise ranking model to lean the image similarity ranking. The aim is to learn an embedding function $f(.)$ that assigns smaller distance to more similar images.

$$D\big(f(p_i), f(p_i^+)\big) < D\big(f(p_i), f(p_i^-)\big)$$
$$\forall p_i, p_i^+, p_i^- \text{ such that } r(p_i, p_i^+) > r(p_i, p_i^-)$$

Where $r\big(p_i, p_j\big)$ is a pairwise relevance score indicating the similarity between to images. They define a hinge loss funcion for a triplet which measures the similarity ranking order for the images p_i^+, p_i, p_i^-. The architecture used by them is has three convnets through which each of the images undergo. At the end of the convnet a feature representation of the image is computed which is then fed to a ranking layer which computes the hinge loss of the triplet. Each of the three convnet has an architecture as shown in figure 3 of [6]. Here they use the same convnet as that of AlexNet [12]. The other two networks in figure 3 of [6] use a shallwer architecture and have less invariance and capture the visual appearance. And finally the outputs from the 3 layers are normalized and combined using a linear embedding layer. For optimizing their model they add an initial layer of sampling which takes care of choosing the most appropriate triplets which need to be fed into the network in order to have a good model training. Without sampling the total possible combination of triplet images is of the order 10^{21}. By doing an intelligent triplet sampling; they significantly reduce this to 24 million triplets which reduces the time significantly. They do the sampling in such a way so as to capture maximum variance in the dataset. The object of this smapling is similar to the hard mining approch done in [5]. They also introduce an efficient online triplet sampling algorithm based on reservoir sampling [27]. They train their model on the ImageNet ILSVRC 2012 dataset [28]. For evaluation they have human raters rate the similarity of a triplet of images. Each human raters is assigned to rate 3 triplets. For an evaluation metric they use two metrics: similarity precision and score at top K for K=30. Similarity precision is defined as the percentage of triplets correctly ranked. Score at top K is defined as the numver of correctly ranked triplets minus the number of incorrectly ranked ones on a subset of triplets whose ranks are higher than K. They compare their proposed approach with existing hand crafted method like Wavelet [29], SIFTlike[15], Fisher [30], HOG [16], SP,Ktexton1024max[31], L1HashKPCA[32], Golden Features and Oasis[33]. They report that their method outperformed the highest of these approaches by 5.4%. They further compare their approach with other convnet based models and show that their approach outperforms the other convnet based appraoches on both the similarity precision and score-at-top-K evaluation metric by 1.1% precision higher than the previous best. In this paper Wang et al. [6] presented a CNN model which incorporates a triplet based hinge based loss ranking function to charaterize fine grained image similarity relationship and a multi-scale neural network to capture global visual features and image semantics. This paper uses a CNN model which is in a way similar to the models used for computing patch similarities in [1][3][4][5][7] and [8].

In this section we presented some of the work relevant to image retrieval systems which uses the basic low level vision tasks on descriptor generation and local image patch matching. Next section focuses on computing depth using CNN models.

IV. Depth Estimation Using CNNs

Here we focus on CNNs being used for depth estimation. One of the fundamental issues in depth from stereo images is local image patch matching to compute the disparity. Various approaches were presented in the previous section for doing the same. A detailed literature review was done by Scharstein and Szeliski [35] in which they detailed all traditional handcrafted methods for computing depth. For computing depth following four steps need to be addressed: 1) Matching Cost computation; 2) Cost aggregation; 3) Disparity Computation / optimization; and 4) Disparity Refinement. Herb [8] presented MC-CNN architecture to compute disparity from a given input patch. They use a convent for replacing the matching cost computation step for computing the disparity. Other 3 steps for computing disparity used were taken directly from state of the art methods and were not a contribution of this paper. The architecture of the convent used consisted of 2 major parts similar to the ones in [3], [4] – a feature extraction part with shared parameters and the feature similarity computation part. The feature extraction part had only one convolutional layer followed by two fully connected layers. For computing the similarity 4 fully connected layers are used followed at the end by a softmax layer. A rectified linear activation is used after every layer except the softmax layer. The network was trained on the KITTI 2012 stereo dataset and performed best. Currently this MC-CNN ranks third on the KITTI stereo benchmark. The top ranking method of displets [36] also uses MC-CNN for the matching cost computation step. One of the major draw-back of MC-CNN is the computational burden of the model making it unsuitable for real world application. This method was the first to take an existing stereo vision pipeline and replace the initial step by using a convnet.

Chen et al. [9] propose a deep visual correspondence embedding model to compute the stereo matching cost to compute disparity. The model is trained by a convnet over the KITTI stereo dataset which has available ground truth disparities. Their proposed model is similar to the siamese based central surround two stream CNN model of [3]. Their proposed architecture can be seen in Fig. 3.

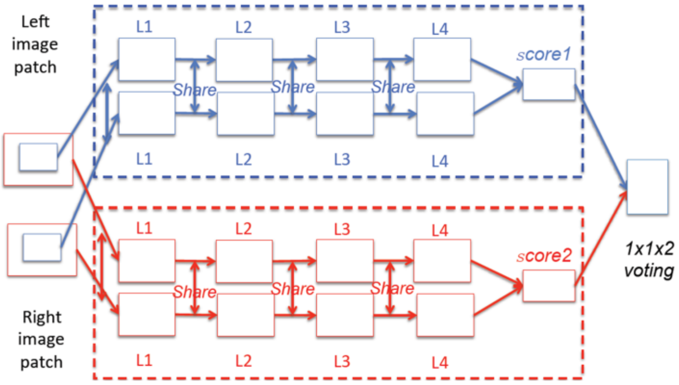


Figure 3. Network architecture of the training model used for deep embedding in [8]. Features are extracted in a pair of patches at different scales followed by an inner product to obtain the matching scores. The scoresfrom different sclaes are then merged for an ensemble.

Their model uses 4 convolutional layers and a similarity score based on dot product. The final feature vector generated by layer 4 of the model is of 200 dimensions. The score S1 and S2 are computed from the 2 models are merged by a 1x1x2 convolutional layer for a weighted ensemble. A deep regression model is applied to minimize the Euclidean cost

$$E(w) = ||S(p,d) - label(p,d)||^2$$

Where $label(p,d) = \{0,1\}$ indicates whether $p^L = (x,y)$ in the left image corresponds to $p - d = (x - d, y)$ in the right image. Their proposed model achieves a 100x speedup at test time compare to the MC-CNN approach of [25]. For computing the final disparity the matching costs generated from the convnet an MRF based stereo framework is used. First a cost volume is initialized by negating the deep embedding scores. The initial costs are then fed to the semi global matcher (SGM) [37] to compute the raw disparity map. After removing unreliable matches via a left-right check, the final disparity map is obtained by propagating reliable disparities to non-reliable areas. The math used for doing the same has been excluded here due to space constraints and can be got from [9]. For evaluation purposes, they compare their obtained results with on the KITTI benchmark dataset against other performers. Currently their proposed approach ranks third on the benchmark. However their model is computationally more efficient than the top 2 approaches. Their approach takes running time of 3 seconds against the 256 and

100 seconds of the first (Displets [36]) and second rank (MC-CNN [25]) approaches. The feature size of their approach is also more compact than the first and second ranked approaches. This suggests the real time feasibility of their approach in the near future as the hardware gets more efficient their approach becomes practically implantable for real world applications like robot navigation, autonomous driving, etc. They also compare their matching costs with the traditional cost matching methods like Census, AD+Gradient, Census+Gradient and the normalized cross correlation approaches and show that their convnet based approach significantly outperforms by a margin of at least 10% on each. For testing the generalizability of their method they also validated their model on the stereo sequences obtained from the Middlebury benchmark [38] using the same CNN model without fine tuning and found their model outperformed the traditional methods significantly.

Estimating depth does not necessarily need a stereo image pair; it can be done from monocular images too. Prediction of depth from stereo images generally implies finding patch correspondences and matching costs as discussed above. However getting depth from monocular images requires integration of both global and local information from an image. Eigen et al. [10] propose a convnet for predicting the depth map from a single image. Their proposed architecture employs two deep network stacks- a course scale network which predicts the depth at a global level and another fine network that refines the global prediction. The Global course scale network is responsible to predict the overall depth map structure using a global view of the scene. The network can be seen in Fig 1 of [10] shows the CNN architecture used by them in detail. The Global Coarse network uses 5 feature extraction convolutional layers followed by 2 fully connected layers. The middle layers combine information from different parts of the image through max-pooling operations to a small spatial dimension. This enables the model to predict the depth by using global information. Next the output of this coarse network is fed to a local fine scale network. This network edits the coarse predictions received from the top model and refines them so as to align it together with local details (objects, walls, line structures). The local fine scale network consists of 4 convolutional followed by a pooling layer. The Fine scale network receives one input one from the initial image and one additional low level feature map from the global coarse network. It integrates the information obtained from each of these and aligns them in a way to predict a refined depth map. In both the networks the hidden layers use Rectified linear activations with the exception of coarse layer 7. A scale invariant error is computed in log scale because it is impossible to recover the exact global scale from a monocular image. However they show if their model gets the mean scale depth from the ground truth they get a 20% relative improvement in the performance. They define the scale-invariant mean squared error (in log space) as:

$$D(y, y^*) = \frac{1}{2n}\sum_{i=1}^{n}(\log y_i - \log y_i^* + \alpha(y, y^*)^2$$

where $y_i$is the predicted depth map, $y_i^*$is the ground truth and $\alpha(y, y^*) = \frac{1}{n}\sum_i (\log y_i - \log y_i^*)$ is the value of αthat minizes the error for a given $(y_{i,}y_i^*)$. For any prediction y, e^{α} is the scale that best aligns it to the ground truth. All scalar multiples of y have the same error, hence the scale invariance. Other than this error metric they also use some other error metrics to validate the performance of their approach. While training they minimize the same scale invariant error as the objective function to be minimized. They do data augmentation by doing transformations like scale, rotation, translation, color, flips to improve their model. Finally they present some results on the NYU Depth v2 [39] and KITTI benchmark [40]. They get better results for not only the scale invariant metric but also the scale dependent metric. This shows that their model not only predicts better relations but also better means. Comparing their approach to the stereo case it is interesting to see that CNNs can predict the depth by integrating the global visual information with the local refinement of it.

V. SUMMARIZATION OF THE PAPERS

Paper #	Description of their approach
Jahrer et al. [1]	Used a Siamese architecture similar to LeCun [18], trained a model to compute a feature representation of a given input patch, use Euclidean distance metric to compute similarity for both SIFT and CNN features and finally show CNN outperforms SIFT
Fisher et al. [2]	Used AlexNet [12] for supervised learning and smaller Net for unsupervised learning, do the task of descriptor matching and show CNN outperform SIFT, Unsupervised learning performs better than supervised but CNNs take more time than SIFT
Zagoruyko and Komodakis [3]	Address the task of local patch matching, use MSER based key point detector [20],propose different CNN models and show that 2 channel based network perform best, similarity function learnt, show CNNs outperform SIFT and Simonyan et al. [24]
Han et al. [4]	Proposed MatchNet CNN model similar to AlexNet [12], compute image patch similarities in way to avoid same patch being sent through network multiple times, similarity function learnt, MatchNet out performs SIFT and [24]
Simo-Serra et al. [5]	Use a Siamese based CNN model using 3 layers, use hard mining strategy to not have network biased to negative class, similarity function is Euclidean norm, feature dimensionality is 128. Outperform SIFT and [24], more efficient than [3],[4]
Wang et al. [6]	Propose a multiscale network based on AlexNet [12], use triplet sampling strategy to rank triplets to perform ranking of images to do image retrieval, similarity metric learnt, outperform traditional methods and CNN based methods
Long et al. [7]	Questions if CNNs good for low level vision, use AlexNet [12] for intraclass alignment, key-point classification and prediction, outperform SIFT flow [21] & NN transfer.
Herb [8]	Do image patch matching to get disparity, propose MC-CNN model to get depth, use best methods for depth pipeline, ranked 3rd on KITTI benchmark at publication time.
Chen et al. [9]	Compute depth from stereo using a mutli scale network model as shown in Fig. 3. Compute the matching cost. Ranked in top 3 on KITTI at publication time. Outperform other traditional cost metrics like Census, AD+Gradient, NCC
Eigen et al. [10]	Compute depth from single image. Propose a CNN model to learn global features which are refined by a fine scale network, compute a scale invariant and scale dependent error which is better than existing methods. Outperform Make3d [46]
Paulin et al. [47]	Use CKNs to learn patch representation in an unsupervised manner using model of PhillipNet [2], do better than others on 4 datasets
Dong and Soatto [34]	Propose a modification to the SIFT algorithm by doing pooling of gradients from different domain sizes. Outperform PhillipNet [2] Convnet by 28.29%. Plot similar graphs as in [2] to prove their point that Traditional Features are still very useful.

VI. CONCLUSION

In this paper we presented a literature survey of papers encompassing tasks relevant for low level vision. We first presented approaches which indicate that CNN based models can extract better feature representation from local image patches, further we showed that one can learn similarity measures and perform better. However these learnt similarity measures may have an inherent bias towards the task at hand. Using a standardized similarity measure is generally more reliable to gauge the representational power of a descriptor. Then tasks like image retrieval and depth estimation were presented as these tasks inherently have to tackle the low level problems as was presented in prior to their description.

It should also be noted that unless one can compute these descriptor in lesser time it does not make much sense for some of the models to be used in real time applications. However with the increasing compute power this would not be too difficult in the near future with the hardware industry coming up with more compute power in the form of powerful GPUs. Also due to some evidence being present from DSP-SIFT [34] work, it should be noted that totally neglecting the traditional approaches may not be a good idea as they too have a good amount of representative power and should be researched in order to find more and more novel ways of computing descriptors in a way to be robust for different tasks. Finally we present some future work ideas in this respect that computing powerful descriptors may be done by carefully integrating Convnet based approaches with traditional approaches.

References

[1] Jahrer, Michael, Michael Grabner, and Horst Bischof. “Learned local descriptors for recognition and matching.” *Computer Vision Winter Workshop*. Vol. 2. 2008.

[2] Fischer, Philipp, Alexey Dosovitskiy, and Thomas Brox. “Descriptor matching with convolutional neural networks: a comparison to sift.” *arXiv preprint arXiv:1405.5769* (2014).

[3] Zagoruyko, Sergey, and Nikos Komodakis. “Learning to compare image patches via convolutional neural networks.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[4] Han, Xufeng, et al. “MatchNet: unifying feature and metric learning for patch-based matching.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[5] Simo-Serra, Edgar, et al. “Discriminative learning of deep convolutional feature point descriptors.” *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[6] Wang, Jiang, et al. “Learning fine-grained image similarity with deep ranking.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[7] Long, Jonathan L., Ning Zhang, and Trevor Darrell. “Do Convnets Learn Correspondence?.” *Advances in Neural Information Processing Systems*. 2014.

[8] Zbontar, Jure, and Yann LeCun. “Computing the stereo matching cost with a convolutional neural network.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[9] Chen, Zhuoyuan, et al. “A Deep Visual Correspondence Embedding Model for Stereo Matching Costs.” *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[10] Eigen, David, Christian Puhrsch, and Rob Fergus. “Depth map prediction from a single image using a multi-scale deep network.” *Advances in neural information processing systems*. 2014.

[11] Girshick, Ross, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[12] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” *Advances in neural information processing systems*. 2012.

[13] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation.” *arXiv preprint arXiv:1511.00561* (2015).

[14] Ji, Shuiwang, et al. “3D convolutional neural networks for human action recognition.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.1 (2013): 221-231.

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

[16] Dalal, Navneet, and Bill Triggs. “Histograms of oriented gradients for human detection.” *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE, 2005.

[17] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool. “Surf: Speeded up robust features.” *Computer vision–ECCV 2006*. Springer Berlin Heidelberg, 2006. 404-417.

[18] LeCun, Yann, et al. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[19] Mikolajczyk, Krystian, et al. “A comparison of affine region detectors.” *International journal of computer vision* 65.1-2 (2005): 43-72.

[20] Matas, Jiri, et al. “Robust wide-baseline stereo from maximally stable extremal regions.” *Image and vision computing* 22.10 (2004): 761-767.

[21] Liu, Ce, Jenny Yuen, and Antonio Torralba. “Sift flow: Dense correspondence across scenes and its applications.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.5 (2011): 978-994.

[22] Brown, Matthew, Gang Hua, and Simon Winder. “Discriminative learning of local image descriptors.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33.1 (2011): 43-57.

[23] Winder, Simon, Gang Hua, and Michael Brown. “Picking the best daisy.” *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[24] Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. “Learning local feature descriptors using convex optimisation.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36.8 (2014): 1573-1585.

[25] Zbontar, Jure, and Yann LeCun. “Computing the stereo matching cost with a convolutional neural network.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[26] Bromley, Jane, et al. “Signature Verification using a “Siamese” Time Delay Neural Network.” In NIPS 1994.

[27] Efraimidis, Pavlos S., and Paul G. Spirakis. “Weighted random sampling with a reservoir.” *Information Processing Letters* 97.5 (2006): 181-185.

[28] Berg, Alex, Jia Deng, and L. Fei-Fei. “Large scale visual recognition challenge 2010.” (2010): 42-55.

[29] Jacobs, Charles E., Adam Finkelstein, and David H. Salesin. “Fast multiresolution image querying.” *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995.

[30] Perronnin, Florent, et al. “Large-scale image retrieval with compressed fisher vectors.” *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[31] Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories.” *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*. Vol. 2. IEEE, 2006.

[32] Ioffe, Sergey. “Improved consistent sampling, weighted minhash and l1 sketching.” *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010.

[33] Chechik, Gal, et al. “Large scale online learning of image similarity through ranking.” *The Journal of Machine Learning Research* 11 (2010): 1109-1135.

[34] Dong, Jingming, and Stefano Soatto. “Domain-size pooling in local descriptors: DSP-SIFT.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[35] Scharstein, Daniel, and Richard Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.” *International journal of computer vision* 47.1-3 (2002): 7-42.

[36] Guney, Fatma, and Andreas Geiger. “Displets: Resolving stereo ambiguities using object knowledge.” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015.

[37] Hirschmüller, Heiko. “Stereo processing by semiglobal matching and mutual information.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 30.2 (2008): 328-341.

[38] Scharstein, Daniel, and Chris Pal. “Learning conditional random fields for stereo.” *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007.

[39] Silberman, Nathan, et al. “Indoor segmentation and support inference from RGBD images.” *Computer Vision–ECCV 2012*. Springer Berlin Heidelberg, 2012. 746-760.

[40] Geiger, Andreas, et al. “Vision meets robotics: The KITTI dataset.” *The International Journal of Robotics Research* (2013): 0278364913491297.

[41] Mairal, Julien, et al. “Convolutional kernel networks.” *Advances in Neural Information Processing Systems*. 2014.

[42] Scale, Krystian Mikolajczyk. “Affine Invariant Interest Point Detectors/Krystian Mikolajczyk and Cordelia Schmid.” *International Journal of Computer Vision* 60.1 (2004): 63-86.

[43] Jégou, Hervé, et al. “Aggregating local descriptors into a compact image representation.” *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010.

[44] Mikolajczyk, Krystian, and Cordelia Schmid. “A performance evaluation of local descriptors.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 27.10 (2005): 1615-1630.

[45] Philbin, James, et al. “Object retrieval with large vocabularies and fast spatial matching.” *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007.

[46] Saxena, Ashutosh, Min Sun, and Andrew Y. Ng. “Make3d: Learning 3d scene structure from a single still image.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31.5 (2009): 824-840.

[47] Paulin, Mattis, et al. “Local convolutional features with unsupervised training for image retrieval.” *Proceedings of the IEEE International Conference on Computer Vision*. 2015.