Automatic Image Annotation using Deep Learning Representations

Venkatesh N. Murthy, Subhransu Majji, R Manmatha
School of Computer Science
University of Massachusetts, Amherst, MA, USA.
{venk,smajji,manmatha}@cs.umass.edu

ABSTRACT

In this paper, we propose simple and effective models for the image annotation task which make use of Convolutional Neural Network (CNN) features extracted from an image and word embedding vectors which represent a tag. Our first model is based on the Canonical Correlation Analysis (CCA) framework that helps in modeling both views - visual feature (CNN feature) and textual feature (word embedding vector) of the data. We report results on all three variants of CCA model namely, linear CCA (CCA), Kernel CCA(KCCA) and CCA with K Nearest Neighbor clustering (CCA-KNN). Among them, our best reported results are using CCA-KNN which outperforms the previously reported results on the Corel-5k and the ESP-Game datasets and yields comparable performance on the IAPRTC-12 dataset. We evaluate the usage of CNN features in the existing models to clearly bring out the advantages of it over dozens of handcrafted feature. We also provide the validation of word embedding vectors with our best performing model (CCA-KNN) in lieu of using binary vectors to represent the tags associated with an image. Our second model is based on CNN, in which we try to find a mapping from input image to word embedding vector by formulating the image annotation problem as a CNN based regression problem (CNN-R).

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and indexing; I.2.10 [**Artificial Intelligence**]: Visual and Scene Understanding

General Terms

Algorithms, Experimentation, Measurement

Keywords

Image annotation, Deep learning, CNN, CCA, Word embeddings

1. INTRODUCTION

Automatic Image annotation is a labelling problem wherein, the task is to predict multiple textual labels for an unseen image describing its contents or visual appearance. Automatic image/video annotation plays an important role in managing the exponentially increasing number of images/videos being uploaded to the internet. For instance, 100 hours of video are uploaded to YouTube every minute and on an average people upload 350 million photos to Facebook each day. Some of this data is partially tagged by the users, but these tags can be ambiguous, incomplete or personalized. Researchers have tried to make use of the metadata associated with the images/videos to build a better classification or object detection system. It has also been used to transfer the tags to unannotated images based on its similarity and vice versa [18, 5, 10, 21, 17]. Our objective is to predict a fixed number of tags for a given test image which accurately describe the visual content.

Most existing techniques are based on Supervised learning, which involves learning a mapping function for low level visual features (color, local descriptors, etc) and high-level semantic concepts (sun, sky, etc). However, the problem of poor annotation (images not being annotated with all relevant keywords) and class-imbalance (large variations in the number of positive samples) makes it a difficult problem to solve. Existing methods use dozens of handcrafted features like quantized Scale Invariant Feature transform (SIFT), quantized color histograms in different color spaces (RGB, LAB, HSV), etc to build a tag prediction model. These models can be generative [5, 17, 32], discriminative [2, 29, 10, 30] or nearest neighbor based and among these, nearest neighbor based models are shown to be the most successful [18, 10, 30].

Multiple features with the right type of model are shown to improve the annotation performance significantly in the current state of the art system [30]. Yet, these dozens of handcrafted features serve as a bottleneck in designing scalable realtime systems. Hence, we propose to use a single CNN feature (representing an image) along with the word embedding vectors (representing the tags associated with it) in our proposed models. CNN features are shown to be successful for most of the vision tasks producing significantly improved results on the most challenging datasets like PASCAL VOC and ILSVRC2013 [6, 24].

Here CNN features are extracted for images using a pre-trained VGG-16 [26] network, and the word embedding vector for a tag is extracted using a pre-trained Skip-gram architecture (Word2Vec) [19]; both these networks are publicly available. The VGG network was designed to classify 1000 classes and it secured both the first and second places in the ISLVRC 2014 challenges for the classification and localization tasks respectively. The skip-gram type of architecture was trained on millions of articles (Google news or Wikipedia) to produce a semantically meaningful real valued vector for any given word. This has found numerous applications in Natural Language processing tasks [28, 3, 27]. This work, to the best of our knowledge, is the first attempt to use CNN features and word embedding vectors to solve the image annotation task and report results on all three standard datasets.

Our proposed models incorporate both CNN features and text features (word embedding vector), one of which is based on the CCA and its variants and the other is CNN based regression. Among all, CCA-KNN significantly outperforms all the previously published results. We are able to achieve this without requiring any computationally expensive metric learning approach as used by almost all successful models [21, 1, 20, 30]. Some early works which used CCA for combing image and text have been proposed [7, 23, 12, 1], but the key differences are, we use CNN features as opposed to multiple handcrafted features (representing images) and we use word embedding vectors instead of binary vectors (representing tags). We demonstrate that this one feature (CNN) performs better than all the multiple handcrafted features used in Ballan et al. [1].

In the CNN based model, the last layer of CaffeNet is replaced with a projection layer (to perform regression) and the resulting network is trained to map images to semantically meaningful word embedding vectors. CaffeNet mainly consists of five convolutional layers and two fully connected layers with some non linear transformation layers. This type of modeling has two advantages: firstly, it does not require dozens of handcrafted features; there is no need for metric learning or ways to combine those features efficiently. Secondly, the approach is substantially simpler to formulate than any other generative or discriminative models. In addition, we also provide the effectivness of CNN features when used in other previously existing models.

2. PREVIOUS WORK

Several algorithms have been proposed for automatic image annotation and retrieval. These can be broadly grouped into three groups - generative models, discriminative models and nearest neighbor based models. Examples of generative models are Cross Media Relevance Model (CMRM) [13], Continuous Relevance Model (CRM) [17] and Multiple Bernoulli Relevance Model (MBRM) [5] . These were the first successful models to solve the image annotation task efficiently. Mixture models can also be treated as a nearest neighbor based approach but mathematically well formulated. In the case of mixture models, the joint probability of words and visual features are determined. Given a test image, this model can be used for computing conditional probability scores for words. Discretized visual features are generally modeled using a non-parametric Gaussian or multinomial distribution and the words are often modeled using multiple Bernoulli distribution. Recently an attempt was made to improve the performance of CRM using Sparse Kernel Learning (SKL-CRM) [1] , in which they try to learn the optimal combination of kernels for the features.

Alternatively, researchers have also proposed discriminative approach for tag predictions such as support vector machine [29], SML [2] , multiple instance learning [8]. These methods build a classifier for each annotation tags by treating them as multi-class multi-labelling problem (either one-versus-all or one-versus-one). Scalability is an issue in this type of setup as it requires pre defined set of models per label.

As mentioned earlier, despite their simplicity nearest neighbor based models have produced state-of-the-art results. In this, the general approach is to predict set of tags for a test image based on some weighted combination of tags present amongst its K neighbor images. Joint Equal Contribution (JEC) model [18] set the baseline for image annotation task using nearest neighbor models, they showed that the equal contribution from different features perform better than computationally expensive L1 regularized logistic regression. TagProp [30] is again based on nearest neighbor model but they achieved significant improvement by using 15 different local and global features along with metric learning. Recently two-pass KNN (2PKNN) technique was proposed [10], which finds semantic neighbors for each test image and the tags are predicted based on the weighted combination of distances. The optimal weights to combine base distances and features was determined via metric learning.

3. PROPOSED METHOD

3.1 Feature extraction

In this section, we provide details about how the CNN features are extracted for images, followed by details about word embedding vectors representing the tags.

3.1.1 CNN features

Given an image, we extract a 4096-dimensional feature vector (X) using a pre-trained CNN on ILSVRC-2012 dataset by VGG team as described in Simonyan et al. [26]. VGG-Net was trained for the image classification task of ImageNet ILSVRC 2014 [25] and secured the first and the second place in the localization and classification tasks respectively. ISLVRC14 contains 1.2 million images which are manually annotated with labels from 1000 words vocabulary. We explored both VGG-16 and VGG-19 layered architecture features. Since both of these gave almost similar performances, we decided to use VGG-16. This choice also helps in reducing the computational complexity. Our initial experimentations involved extracting features from BVLC-Net provided by Caffe [14] which is similar to Krizhevsky Net [16] but its performance was comparatively lower than VGG-16 features, hence we decided to utilize VGG-16 features for all our experiments. VGG-16 Features are computed by forward propagating a mean-subtracted 224x224 RGB image through eight convolutional layers and three fully connected layers. In our case, we resize all the images irrespective of their aspect ratio to 224x224 to make it compatible with pre-trained CNN (VGG-16).

3.1.2 Word embeddings

For each tag associated with an image, we represent the tag (word) by a 300 dimensional real valued feature vector (semantically meaningful) using a Word2Vec tool and we name it as a word embedding vector ($W \in \mathbb{R}^{lxq}$), where l is the number of labels and $q = 300$ dimensions. These word vectors are obtained from pre-trained skip-gram text modeling architecture introduced by Mikolov et al. [19]. The network was trained on Google News dataset, partly (about 100 billion words). It was shown that the model learns similar embedding vectors for semantically related words, therefore we make use of it to represent the annotation tags. In our case, we take an average of all the word embedding vectors (Y) to represent an image since they are associated with multiple tags . Formally, if there are k tags associated with an image I then

$$Y = \frac{1}{k}\sum_{i=1}^{k} W_i \tag{1}$$

and their association is represented as $\{I, Y\}$.

We also tried treating each image with its respective word embedding vectors as being completely independent. For example, image I is represented by word embedding vectors $\{W_i, W_j \ and \ W_k\}$, then they are represented as $\{I, W_i\}$, $\{I, W_j\}$ and $\{I, W_k\}$ to model it in CCA. This type of representation was found to yield similar performance and was also computationally expensive when compared to averaging the word embedding vectors. Henceforth, we chose to average the embedding vectors instead of treating them independently. While reporting the result we refer to word embedding vectors as W2V.

3.2 CCA based model

3.2.1 Canonical Correlation Analysis (CCA)

Given a pair of views for an image - one being Visual feature (X, CNN feature) and the other being textual feature (Y, word embedding vector), we try to obtain optimized projections w_x and w_y for X and Y respectively. CCA helps in finding the two basis ($w_x \, and \, w_y$) such that the correlation between the projected representations are maximized. The dimensionality of these new basis vectors is equal to or less than the smallest dimensionality of the two variables. An important property of canonical correlations is that they are invariant with respect to affine transformations of the variables. Mathematically, for M samples, let $X \in \mathbb{R}^{mxp}$ and $Y \in \mathbb{R}^{mxq}$ be the two views of the data, then the optimized projection vectors $w_x \ and \ w_y$ are computed such that the following correlation coefficient ρ is maximized.

$$\rho = \underset{w_x, w_y}{\arg\max} \frac{w_x^T XY^T w_y}{\sqrt{(w_x^T XX^T w_x)(w_y^T YY^T w_y)}} \tag{2}$$

The solution can be found by formulating it as a generalized eigen value problem [11]:

$$XY^T(YY^T)^{-1}YX^T w_x = \eta XX^T w_x \tag{3}$$

where η is the eigenvalue corresponding to the eigenvector w_x. Multiple such projectors can be found to form a projection matrix $W_x \in \mathbb{R}^{pxl} \in$ and similarly $W_y \in \mathbb{R}^{qxl} \in$ can also be determined. In the case of regularized CCA, a regularization term λI with $\lambda > 0$ is added to XX^T in Eq. 3 to avoid overfitting.

3.2.2 Kernel Canonical Correlation Analysis (KCCA)

Since CCA can only capture linear relationships, we propose to use a Chi-squared kernel for exploiting the non linear relationships. Experimentally chi-square was found to be well suited for this kind of experiments. In this case, the visual feature X is virtually mapped into a high dimensional feature space $\mathcal{H}_x$ using a mapping function ϕ_x. The ϕ_x mapping can be achieved by a positive definite kernel function $K_x = \langle \phi_x, \phi_x \rangle \in \mathbb{R}^{mxm}$, where $\langle :, : \rangle$ is an inner product in $\mathcal{H}_x$. Similarly, we can map the word embedding vector Y to $\mathcal{H}_y$ space using the kernel function $K_y = \langle \phi_y, \phi_y \rangle \in \mathbb{R}^{mxm}$. Kernel CCA tries to find the solution of w_x and w_y as a linear combination of the training data:

$$w_x = \phi_x \boldsymbol{\alpha} = \sum_{i=1}^{m} \alpha_i \phi_x(x_i) \tag{4}$$

$$w_y = \phi_y \boldsymbol{\beta} = \sum_{i=1}^{m} \beta_i \phi_y(y_i) \tag{5}$$

The vector w_x and w_y can be determined by maximizing:

$$\boldsymbol{\alpha}, \boldsymbol{\beta} = \underset{\boldsymbol{\alpha}, \boldsymbol{\beta}}{\arg\max} \frac{\boldsymbol{\alpha}^T K_x K_y \boldsymbol{\beta}}{\sqrt{(\boldsymbol{\alpha}^T K_x^2 \boldsymbol{\alpha})(\boldsymbol{\beta}^T K_y^2 \boldsymbol{\beta})}} \tag{6}$$

Since feature vector dimensions were high, we encountered the problem of overfitting. In order to avoid this we used regularized Kernel CCA which finds $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ by maximizing the following objective function:

$$\underset{\boldsymbol{\alpha}, \boldsymbol{\beta}}{\arg\max} \frac{\boldsymbol{\alpha}^T K_x K_y \boldsymbol{\beta}}{\sqrt{(\boldsymbol{\alpha}^T K_x^2 \boldsymbol{\alpha} + r_x \boldsymbol{\alpha}^T K_x \boldsymbol{\alpha})(\boldsymbol{\beta}^T K_y^2 \boldsymbol{\beta} + r_y \boldsymbol{\beta}^T K_y \boldsymbol{\beta})}} \tag{7}$$

The solution to this problem yields top l eigen vectors $\mathcal{W}_x = [\boldsymbol{\alpha}^1 \ldots \boldsymbol{\alpha}^l]$ and $\mathcal{W}_y = [\boldsymbol{\beta}^1 \ldots \boldsymbol{\beta}^l]$ which forms a projection matrix.

3.2.3 Implementation details

CCA and KCCA with regularization was implemented as explained in [11]. Regularization played an important role to avoid overfitting which in turn resulted in better performance. In the case of linear CCA, we project X onto W_x, project Y onto W_y and project W onto W_y:

$$U = (X - \mu_X) * W_x \tag{8}$$

$$V = (Y - \mu_Y) * W_y \tag{9}$$

$$Z = W * W_y \tag{10}$$

Given a test image I_t, we extract deep learning visual features V_t and project it using w_x as follows:

$$T = (V_t - \mu_X) * W_x \tag{11}$$

and compute the correlation distance to V. The corresponding tags associated with the closest matching V_i are assigned to the test image (tags are also ranked according to their frequency in the training dataset). If the tags are less than the fixed annotation length then we pick the next closest match and transfer the tags, we repeat this until we obtain required set of tags, in our case its five (because to have fair comparison with previous work).

Similarly, in the case of KCCA, we kernelize X, Y and Z and later project onto $\mathcal{W}_x, \mathcal{W}_y, \mathcal{W}_x$ respectively. For a test image, we kernelize the visual features and follow the same procedure as above.

In CCA with KNN clustering (CCA-KNN) setup, after finding the correlation distance of T with V, we choose K semantic neighbor samples from each cluster (grouped according to its labels) for that particular test image and now their associated tags forms a subset of tags Z_k (potential candidates for a test image). Later, we rank the words w for a test image I_t according to its probability score of:

$$P(I_t|w) = \sum_k exp(-D(T, Z_k))\mathbb{1}_k(w) \tag{12}$$

where, $D(T, Z_k)$ is the correlation distance between T and Z_k and $\mathbb{1}_k(w)$ is an indicator function which takes a value 1

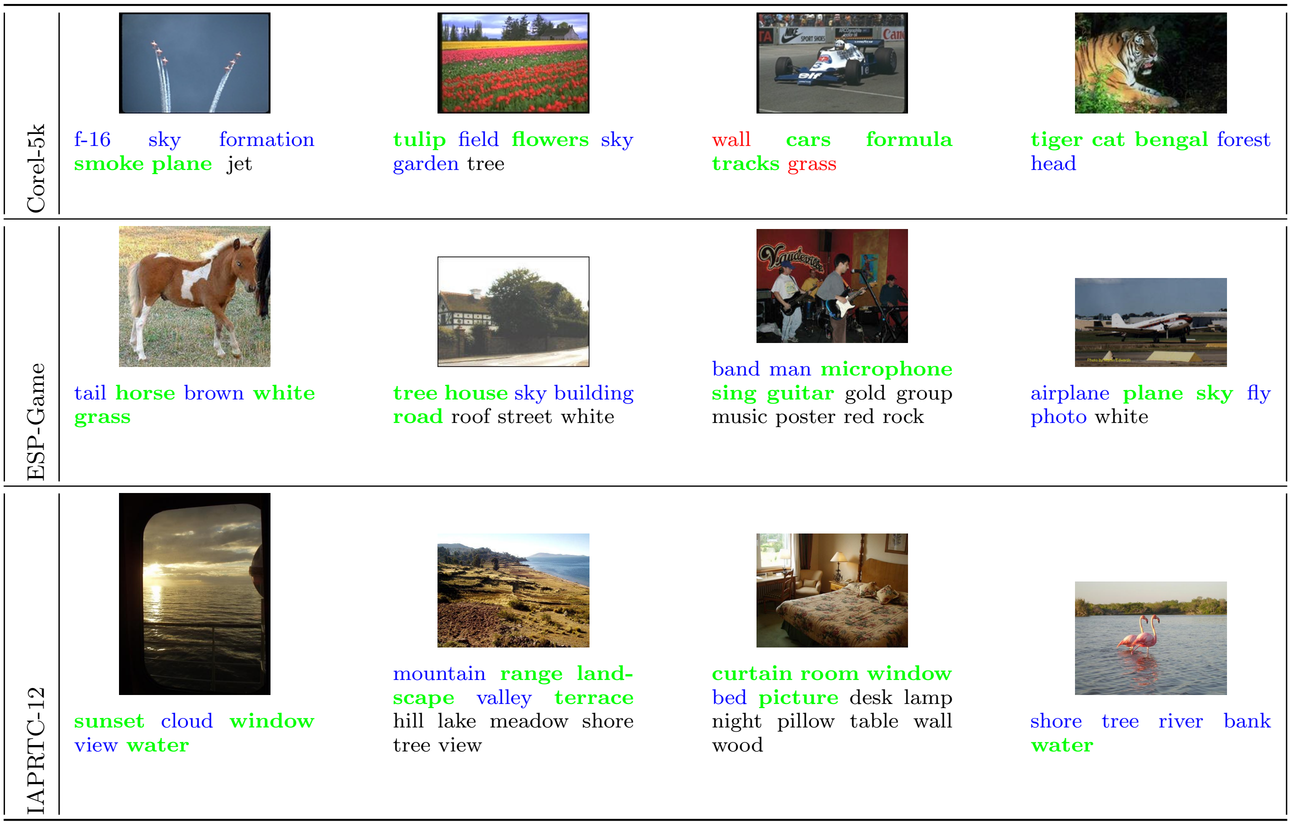

Corel-5k
f-16 sky formation smoke plane jet
tulip field flowers sky garden tree
wall cars formula tracks grass
tiger cat bengal forest head
ESP-Game
tail horse brown white grass
tree house sky building road roof street white
band man microphone sing guitar gold group music poster red rock
airplane plane sky fly photo white
IAPRTC-12
sunset cloud window view water
mountain range landscape valley terrace hill lake meadow shore tree view
curtain room window bed picture desk lamp night pillow table wall wood
shore tree river bank water

Figure 1: examples of randomly sampled images which are automatically annotated with CCA-KNN model. First row: Corel-5k, second row: ESP-Game and third row: IAPRTC-12 datasets.

if the tag is present among neighbors and 0 otherwise. This model was inspired by Verma et al. [30] and Guillaumin et al. [10].

3.3 CNN based regression model

Inspired by the success of deep CNN architectures [16, 26, 6] on the large scale image classification task [25] we intend to make use of this powerful architecture to solve the task of automatic image annotation. To the best of our knowledge, this is the first attempt to formulate this problem based on CNN, which is simple yet powerful.

The idea is to formulate the problem as a linear regression. We achieve this by replacing the last layer of BVLC Net with a projection layer (fully connected layer) and we call it as a CNN regressor (CNN-R). CNN provides the mapping function which regresses the fixed size of the input image to a word embedding vector. In detail, the network consisted of five convolutional layers and two fully connected layers with some series of non-linear transformation (rectified linear unit) and pooling layers. Most importantly, it had some dropout layers in addition, to avoid overfitting. For further details, please refer to [16]. In this setup, we increased the learning rate for the newly introduced layer while reducing it for all the other previous layers and the reason being that we are trying to fine-tune the network previously trained on 1.2 million images. The input image size was fixed to be 227x227 and the final regressed output was a 300 dimensional vector. Since we have chosen to do linear regression, we use Euclidean loss (L2) instead of Softmax loss during the training phase. The prediction layer tries to predict the word embedding vector by minimizing the L2 loss depending on which, the model parameters are updated using Back propagation algorithm. The training and the testing network architectures are provided in the appendix.

4. EXPERIMENTS

4.1 Datasets

We evaluate on three standard publicly available image annotation datasets - Corel-5k, ESP-Game, IAPRTC-12. These datasets contain a variety of images like natural scene, game, sketches, transportation vehicles, personal photos and so on, thus making it a challenging task.

Corel 5k: The dataset consists of 5000 images, among which 4500 are used for training and the rest 500 images are used for testing [4]. The vocabulary consisted of 260 tags used for image annotation. Each image was annotated with varying number of tags from 1 to 5 and had an average of 3.5.

ESP Game: It consists of 20,770 images in total. Images are annotated via an online gaming setup [18]. If the images are annotated with the same key words by two distinct players, then they score a point. The training dataset consists of 18,689 images and the test set consists of 2081 images. Image annotation vocabulary consists of 268 tags and on an average each image was annotated with 4.7 tags.

IAPRTC-12: It is a collection of 19,627 images of natural scenes which are split into 17665 training set and 1962 test-

	Feature		Corel-5K				ESP Game				IAPRTC-12			
Method	Visual	text	P	R	F	N+	P	R	F	N+	P	R	F	N+
CRM [17]	HC	-	16	19	17	107	-	-	-	-	-	-	-	-
SML [2]	HC	-	23	29	26	137	-	-	-	-	-	-	-	-
MRFA [32]	HC	-	31	36	33	172	-	-	-	-	-	-	-	-
GS [33]	HC	-	30	33	31	146	-	-	-	-	32	29	30	252
JEC [18]	HC	-	27	32	29	139	22	25	23	224	28	29	29	250
CCD [22]	HC	-	36	41	38	159	36	24	29	232	44	29	35	251
KSVM-VT [29]	HC	-	32	42	36	179	33	32	33	259	47	29	36	268
MBRM [5]	HC	-	24	25	25	122	18	19	19	209	24	23	24	223
TagProp(σML) [10]	HC	-	33	42	37	160	39	27	32	239	46	35	40	266
2PKNN+ML [30]	HC	-	44	46	45	191	53	27	36	252	54	37	**44**	278
SVM-DMBRM [21]	HC	-	36	48	41	197	**55**	25	34	259	56	29	38	**283**
KCCA-2PKNN [1]	HC	-	42	46	44	179	-	-	-	-	**59**	30	40	259
SKL-CRM [20]	HC	-	39	46	42	184	41	26	32	248	47	32	38	274
JEC	VGG-16	-	31	32	32	141	26	22	24	234	28	21	24	237
2PKNN	VGG-16	-	36	49	44	198	37	33	35	254	50	35	39	279
SVM-DMBRM	VGG-16	-	42	45	43	186	51	26	35	251	58	27	37	268
CCA	DL	W2V	35	46	40	172	29	32	30	250	33	32	33	268
KCCA	DL	W2V	39	**53**	45	184	30	36	33	252	38	39	38	273
CCA-KNN	DL	W2V	**42**	52	**46**	**201**	46	**36**	**41**	**260**	45	**38**	41	278
CNN-R	DL	W2V	32	41.3	37.2	166	44.5	28.5	34.7	248	49	31	37.9	272

Table 1: Experimental results of our proposed models with previously reported best scores on three datasets; Corel-5K, ESP Game, and IAPRTC-12. P: Average Precision, R: Average Recall, N+: Number of distinct words that are correctly assigned to at least one test image. For all of the numbers the higher the better.

	Feature		Corel-5K				ESP Game				IAPRTC-12			
Method	Visual	text	P	R	F	N+	P	R	F	N+	P	R	F	N+
CCA-KNN	DL	BV	39	51	44	192	44	32	37	254	41	34	37	273
CCA-KNN	DL	W2V	42	52	46	201	46	36	41	260	45	38	41	278

Table 2: Experimental results of CCA-KNN proposed model to show the effectiveness of word embedding vectors.

ing set [9]. Vocabulary consisted of 291 tags with an average of 5.7 tags used for annotating each image.

4.2 Evaluation Measure

Most papers use one metric. There is a small number of papers which use other metrics. Hence, it is important to clearly state all three variations in the evaluation type. Firstly, we present the most widely reported type of evaluation where the recall and precision are computed **per word** and their average over all the words are reported [21, 1, 20, 30, 18, 5, 17]. In the second type of evaluation, the precision and recall are computed **per image** and their average over all the test images are reported [31]. In the third type of evaluation, precision and recall are computed **per word** but they are computed only for **non-zero recall** words and their average over all non-zero recall words are reported [15]. We strictly adhere to computing the recall and precision **per word** (for all the words) and reporting their means over all the words, thus making it a fair comparison to the majority of the works in this area. Along with that, we also report **N+** denoting the number of words with non-zero recall value. Since we follow the fixed length annotations of 5 words per test image for all three datasets, we will never be able to achieve perfect precision and recall.

4.3 Experimental Results and Discussion

Experimental results are reported on three standard image annotation datasets; Corel-5k, ESP-Game and IAPRTC-12. In order to have a fair comparison with the previously reported results, we follow the same train and test split as reported in [10] and also fix the length of the annotations (five tags) for a test image. In the first subsection, we evaluate the CNN features with our proposed model and compare its performance with all other previous work. In addition, we also provide results of using some of the existing models like 2PKNN, JEC and SVM-DMBRM with the new set of CNN features. This helps in understanding how well the CNN features perform against 15 handcrafted features (local + global).

In the second subsection, we evaluate the importance of the word embedding vectors. We choose to do this by picking the best performing model in the Table 1 and use word embedding vectors instead of frequency counts to represent the tags. Since each image is annotated with a unique set of tags in all three datasets, the frequency vector repressing the tags of an image just turns out be a simple binary vector (BV).

4.3.1 Evaluation of CNN features

Results are provided in the Table 1. We can clearly see that our proposed CCA-KNN model outperforms all other previous work on Corel-5k and ESP-Game datasets. More precisely, our method provides 2.6% and 13.5% increase in F1 measure on Corel-5k and ESP-Game datasets respectively. Also we are better in terms of N+ (# non-zero recall) measure, which is a clear indication that our method provides a generalization for unseen test images. In the case of IAPRTC-12 dataset, our model yields comparable results to the state-of-the-art. One possible reason for not getting a significant performance improvement on IAPRTC-12 is that, it consists of lot of variations in the number of tags annotated per image. The mean, median and maximum number of labels per image in this dataset are 5.7, 5 and 23 respectively and since there is a significant difference in the mean (or median) and the maximum number of labels per image it clearly indicates a poor annotation or weak labelling.

Most importantly, CNN features with state-of-the-art method 2PKNN model yield results close to their originally reported results using 15 features with metric learning. Also, the usage of CNN features on some of the recently proposed models like SVM-DMBRM and KCCA-2PKNN gives almost similar performance compared to their originally published results. This clearly indicates that using CNN features over 15 handcrafted features can help in reducing the computational complexity and time (extracting 15 features and metric learning) to a great extent without much compromise on the performance. This becomes very helpful in real world applications like web search and retrieval.

4.3.2 Importance of word embeddings

Table 2 provides the results of using word embedding vectors (W2V) over binary vectors (BV) with our best performing model CCA-KNN. It is clearly seen that word embedding vector along with CNN features within CCA-KNN model yields the best performance on all three datasets. This suggests that word embedding vectors provide better representation for words than its binary form. This could be because semantically related words tend to have similar word embedding vectors. For example, France equals France (cosine distance of 1), while Spain has a cosine distance of 0.678515 from France, the highest of any other country in the word embeddings space.

4.3.3 Qualitative analysis

In Figure 1 we provide some examples of randomly sampled images from all three datasets. These images are all automatically annotated with CCA-KNN model. The tags in green (bold) are correctly matched with the groundtruth, tags, marked in blue (italics) are the semantically meaningful ones missing in the groundtruth, the tags marked in black (normal text) are the ones which our model failed to predict because of the fixed annotation length restriction and tags in red color are predicted incorrectly by our model. We can clearly see that some images are poorly annotated (missing tags) but still our method is able to retrieve those semantically meaningful tags. In the other case, since we are restricted to a fixed length of annotation (five per image), our model might miss some tags present in the groundtruth.

4.3.4 Evaluation of CNN-R model

Experimental evaluation results of our CNN-R models on all three datasets are provided in Table 1 with comparison to some of the best performing models in the literature. From Table 1, we can clearly observe that CNN-R outperforms most of the models and gives competitive performance to the current state-of-the-art 2PKNN model. This shows that CNN-R has a clear advantage over all the existing methods because of the following reasons: no need to extract multiple low level features and to incorporate high level semantics; no metric learning; we can fine-tune the deep architecture even for a small dataset and this type of model is also capable of predicting new set of previously unseen classes with the help of word embeddings vectors.

We believe that the performance can be improved further with some regularization.

5. CONCLUSION

We have explored the CNN features and word embedding vectors for the image annotation task and have introduced some new models to take advantage of both these features. Empirically, we showed that one of our proposed model CCA-KNN yields significantly better results when compared to all other previous work. In addition, we validated the advantage of using CNN features over 15 handcrafted features in some of the existing models and showed that the performance are comparable to their previously reported results. All of our proposed models clearly have an advantage of not having to compute those multiple engineered features and later figuring out the optimal weights to combine them with the help of some computationally expensive metric learning techniques (this has been the current trend in almost all of the recent works). Further, we introduced a simple and efficient way of formulating the image annotation problem as CNN based regressor which could be very useful in the real world applications. Moving forward, we will be experimenting with replacing the last year of CaffeNet with a CCA layer instead of regression to further improve the performance.

6. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] L. Ballan, T. Uricchio, L. Seidenari, and A. Del Bimbo. A cross-media model for automatic image annotation. In *Proceedings of International Conference on Multimedia Retrieval*, page 73. ACM, 2014.

[2] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):394–410, Mar. 2007.

[3] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[4] P. Duygulu, K. Barnard, J. F. G. d. Freitas, and D. A. Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *Proceedings of the 7th European Conference on*

Computer Vision-Part IV, ECCV '02, pages 97–112, London, UK, UK, 2002. Springer-Verlag.

[5] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR'04, pages 1002–1009, 2004.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[7] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *International Journal of Computer Vision*, 106(2):210–233, 2014.

[8] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(8):1371–1384, 2008.

[9] M. Grubinger. *Analysis and evaluation of visual information systems performance*. PhD thesis, Victoria University, 2007.

[10] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *In ICCV*, 2009.

[11] D. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[12] D. R. Hardoon and J. Shawe-Taylor. Kcca for different level precision in content-based image retrieval.

[13] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 119–126, 2003.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[15] M. M. Kalayeh, H. Idrees, and M. Shah. Nmf-knn: Image annotation using weighted multi-view non-negative matrix factorization. June 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *in NIPS*. MIT Press, 2003.

[18] A. Makadia, V. Pavlovic, and S. Kumar. A new baseline for image annotation. In *Proceedings of the 10th European Conference on Computer Vision: Part III*, ECCV '08, pages 316–329, Berlin, Heidelberg, 2008. Springer-Verlag.

[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] S. Moran and V. Lavrenko. A sparse kernel relevance model for automatic image annotation. *International Journal of Multimedia Information Retrieval*, 3(4):209–229, 2014.

[21] V. N. Murthy, E. F. Can, and R. Manmatha. A hybrid model for automatic image annotation. In *Proceedings of International Conference on Multimedia Retrieval*, ICMR '14, pages 369:369–369:376, New York, NY, USA, 2014. ACM.

[22] H. Nakayama. *Linear distance metric Learning for large-scale generic image recognition*. PhD thesis, The University of Tokyo, Japan, 2011.

[23] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *Proceedings of the international conference on Multimedia*, pages 251–260. ACM, 2010.

[24] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014.

[25] 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, 2014.

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics, 2011.

[28] J. Turian, L. Ratinov, and Y. Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[29] Y. Verma and C. Jawahar. Exploring svm for image annotation in presence of confusing labels. In *Proceedings of the 24th British Machine Vision Conference*, 2013.

[30] Y. Verma and C. V. Jawahar. Image annotation using metric learning in semantic neighborhoods. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12, pages 836–849, Berlin, Heidelberg, 2012. Springer-Verlag.

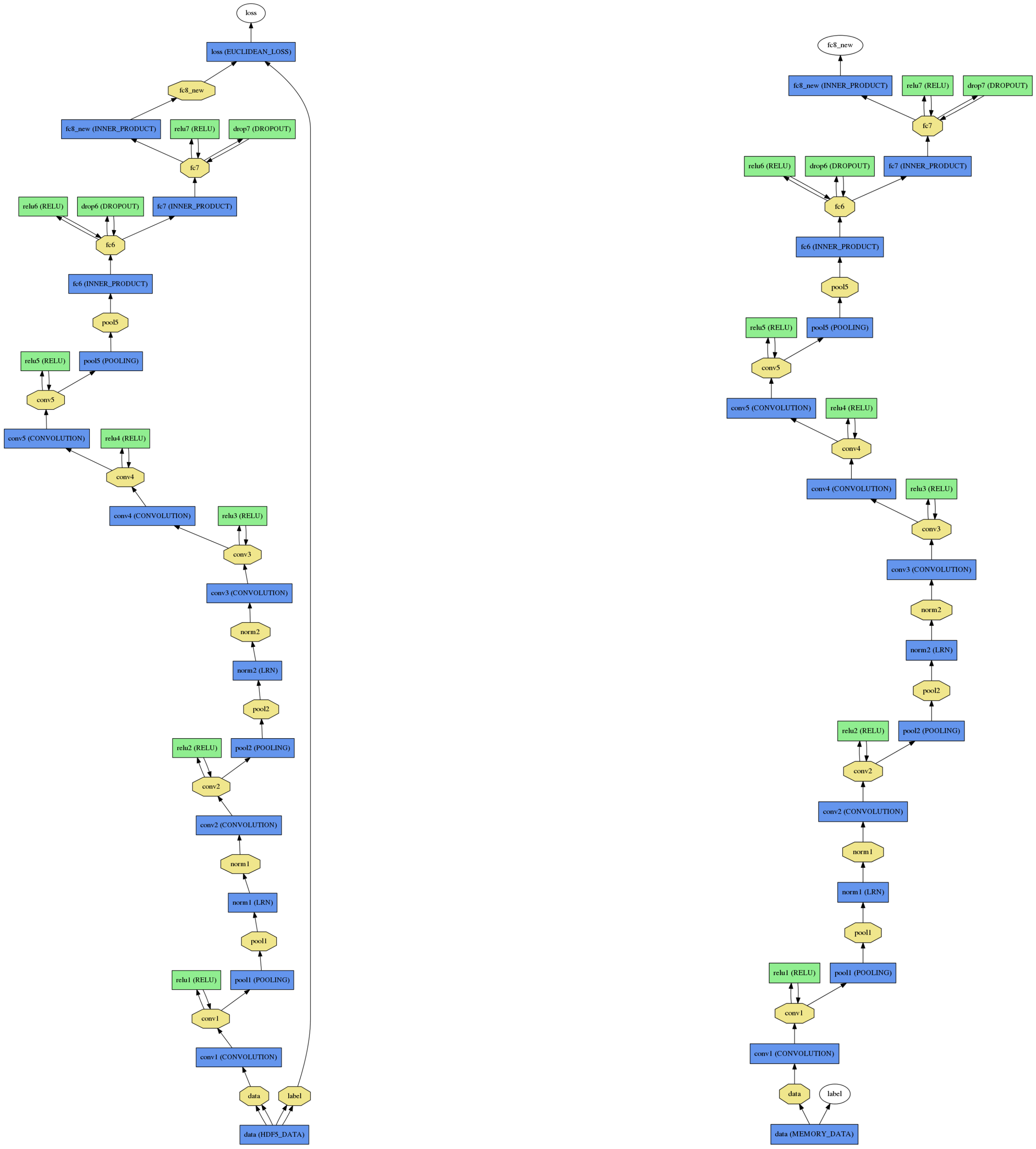
[31] L. Wu, R. Jin, and A. K. Jain. Tag completion for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(3):716–727, 2013.

[32] Y. Xiang, X. Zhou, T. Chua, and C. Ngo. A revisit of generative models for automatic image annotation using markov random fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[33] S. Zhang, J. Huang, Y. Huang, Y. Yu, H. Li, and D. Metaxas. Automatic image annotation using group sparsity. In *IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), 2010.

APPENDIX



(a) Training architecture

(b) Testing architecture

Figure 2: Details of CNN-R architecture