Electronic Letters on Computer Vision and Image Analysis 21(1):1-16, 2022

Pre-trained CNNs as Feature-Extraction Modules for Image Captioning: An Experimental Study

Muhammad Abdelhadie Al-Malla*, Assef Jafar* and Nada Ghneim+

* *Informatics Department, Higher Institute for Applied Sciences and Technology, Masaken Barzeh, Damascus, Syria*
+ *Department of Informatics and Communication Engineering, Arab International University, Ghabagheb, Daraa, Syria*

Received 6th of May, 2021; accepted 30th of March 2022

Abstract

Many recent image captioning works employ the Encoder-Decoder architecture, with Convolutional Neural Networks (CNNs) as feature extractors. This work presents a thorough experimental study about feature extraction using CNNs for the task of image captioning, in the context of deep learning. We examined 12 feature extraction architectures (from the VGG, ResNet, Inception, InceptionResNet, DenseNet, and NASNetLarge model families) and assessed their effectiveness as feature extractors using image captioning quality measures. The total is 72 experiments on 12 image classification CNNs, pre-trained on the ImageNet dataset. The features are extracted from the last layer after removing the fully connected layer and fed into the captioning model. We used a unified captioning model with a fixed vocabulary size across all the experiments to study the effect of changing the CNN feature extractor on image captioning quality. The scores are calculated using the standard metrics in image captioning. We found a strong relationship between the CNN model structure and the image captioning dataset, and that among the tested feature extraction CNNs, Xception and InceptionResNet V2 were the most robust while the two VGG models gave the least quality for image captioning. Based on these results, we recommend a set of pre-trained CNNs for each of the image captioning evaluation metrics that we want to optimise. To our knowledge, this work is the most comprehensive comparison between feature extractors for image captioning.

Key Words: Convolutional Neural Network, Feature Extraction, Image Captioning, Deep Learning.

1 Introduction

Image captioning is one of the trending problems in modern Artificial Intelligence (AI). It is concerned with generating an output text describing an input image, where the output can be one or more sentences. Image captioning crosses the fields of computer vision and natural language processing. The problem of image captioning is traditionally solved using machine learning techniques, and recently deep learning techniques are gaining more popularity for such applications.

Correspondence to: <abdelhadie.almalla@hiast.edu.sy>

Recommended for acceptance by $AngelD.Sappa$
https://doi.org/10.5565/rev/elcvia.1436
ELCVIA ISSN:1577-5097
Published by Computer Vision Center / Universitat Autònoma de Barcelona, Barcelona, Spain

For a long time, in computer vision problems, features had to be extracted by human-engineered feature extraction algorithms, such as the Scale-Invariant Feature Transform (SIFT) proposed by Lowe [1] and the Speeded-Up Robust Features (SURF) proposed by Bay et al. [2]. However, with the advancements in convolutional neural networks and the good results achieved when letting CNNs automatically discover the features in order to classify images, the automatic feature extraction done by the convolutional layers in a CNN is receiving more attention.

This superiority of CNN features for high-level vision tasks has been demonstrated empirically in many previous works. In this regard, Gong et al. [3] presented an analysis of Content-Based Image Retrieval (CBIR) methods, in which CNN methods yielded better performance in the overall-precision and overall-recall metrics. Furthermore, Shin et al. [4] did a multi-class sentiment image classification experiment to choose their feature extraction method for sentimental image captioning. When they used features extracted from the VGG model, they obtained much better Top-1 accuracy of classification on the Sentiment dataset than the accuracy obtained when they used SIFT features combined with a global color histogram.

Previous works in this domain handled image representation from different aspects, but did not investigate the direct effect of the feature extraction model on image captioning quality measures. The success of some CNNs in a domain does not necessarily imply that they are going to succeed in another. This suggests the need for a study that investigates image representation models for the task of image captioning.

In this work, we explore how well different pre-trained CNNs perform when used as the feature extraction module of an image captioning system. We examine the performance of 12 pre-trained CNNs, trained on the ImageNet dataset [5], in an image captioning system that uses soft attention and an Encoder-Decoder architecture. The experiments were done on three standard benchmark datasets in the image captioning field: Flickr8k [6], Flickr30k [7] and MS COCO [8]. Figure 1 depicts the phases of image captioning in deep learning. In this work, we focus on the CNN visual feature extraction phase.

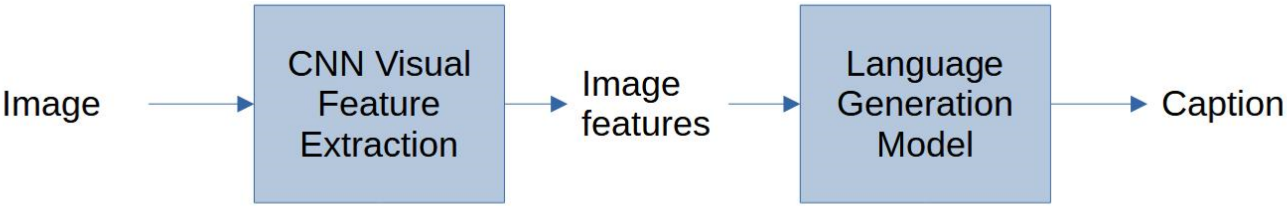


Figure 1: An illustration of image captioning in deep learning.

In the rest of this paper, we discuss in section 2 the related works in the domain. In section 3, we present our methodology, which includes the adopted network architectures, the datasets we used, and the pre-processing that we performed. In section 4, we present the design and the results of our experiments, compared to previous works. In section 5, we conclude our work.

2 Related Works

Typical CNNs consist of convolutional layers, pooling layers and fully connected layers, with the output of each layer being a function of the output of previous layers. In order for a CNN to be used in image classification, its output must be invariant to semantically-irrelevant changes. Several recent works tried to unravel the power of CNNs in feature extraction, such as the works in [9], [10] and [11] that experimented with features from the output of multiple layers of the network.

In [12], Tran et al. worked on solving the problem of describing images in the wild. They addressed the challenges of captioning quality with respect to human evaluation, handling out-of-domain data, and low latency. The model can detect a wide range of visual concepts. Their work included developing an entity recognition model to identify celebrities and landmarks, and a confidence model for caption output. For feature extraction, they used ResNets.

In [13], Valev et al. made a comparison of state-of-the-art pre-trained models on fine-grained image classification using the Stanford Cars-196 dataset [14]. Interestingly, the top two accurate methods (DenseNet161 and DenseNet121) were the two recommended pre-trained CNNs in [15] as feature extraction modules.

In [16], a method for scaling was introduced to scale the depth, width and resolution uniformly. The authors demonstrated the usefulness of this method on scaling up MobileNets and ResNets. They also included a comparison of state-of-the-art pre-trained CNNs in image classification. In [17], Xie et al. presented a comparison with more focus on their Noisy Student Training method.

In [18], Irvin et al. performed an experiment to test the performance of ResNet152, DenseNet121, Inception V4, and SEResNeXt101 on the CheXpert dataset[*], with DenseNet121 performing the best. In [19], DenseNets were used in nine of the top ten CheXpert competition models as a part of their ensemble, while DenseNets were overperformed on ImageNet.

In [15], Holliday and Dudek performed a wide-range evaluation of CNNs as feature extractors for matching visual features under large changes in appearance, perspective and visual scale. Their evaluation covers 82 different layers from twelve different CNN architectures belonging to four families: AlexNets, VGG Nets, ResNets and DenseNets, evaluating their usefulness in matching tasks under challenging variations in perspective and appearance. They found significant differences both in robustness and feature size among different architectures. According to their work, the overall best features were the outputs of the third transition block of DenseNet architectures, especially DenseNet121 and DenseNet161, which provide slightly different trade-offs of accuracy to feature size.

Most image captioning evaluation metrics over-penalise mismatches between reference and generative captions because of not considering the intrinsic variance between ground truth captions. In [20], Yi et al. introduced a novel metric based on the metric of BERTScore to handle this challenge. It extends the BERTScore with features appropriate for image captioning.

The work of Sharif et al. [21] contained a comparison of six CNNs as global feature extractors for their image captioning model. They tested the Inception V3, DenseNet201, InceptionResNet V2, ResNet152 V2, Xception and NASNet CNNs. They used Flickr30k for testing. NASNet gave the best results in their experiment.

In [22], Zhang et al. proposed an image captioning model with a variation of the Long Short-Term Memory (LSTM) called parallel-fusion LSTM (pLSTM). It fuses two LSTM units by the hidden state at each time step. This makes the attributes and visual information complementary for generating more accurate descriptions. The first variation is pLSTM with attention (pLSTM-A), which captures the crucial semantic and visual information for generating captions. The second variation (pLSTM-G) directly adjusts the hidden state of a visual LSTM using synchronous semantic information to the critical region.

[*] CheXpert is a dataset of medical chest X-ray images

In [23], Zhang et al. incorporated the Transformer model for the task of image captioning. They improved the Transformer model in two manners. The first is augmenting the Maximum Likelihood Estimation (MLE) with an extra Kullback-Leibler (KL) divergence term to distinguish the difference between incorrect predictions. The second is a method that they introduced for leveraging the knowledge graph to help the Transformer to generate captions.

In [24], Ke et al. investigated the feature extraction performance of 16 popular CNNs on CheXpert. They did not find a relationship between the performance on ImageNet and the performance on the medical image dataset. However, they found out that the choice of CNN architecture influences performance more than the concrete model within the model family for medical tasks. They also noticed that pre-training on ImageNet gives a boost to performance in all architectures, with a lower boost for bigger ones.

3 Our Approach

In this paper, we compare 12 architectures (from the VGG, ResNet, Inception, InceptionResNet, DenseNet, and NASNetLarge model families) for feature extraction and evaluate their performance as feature extractors using the metrics used for image captioning. The feature extraction model was incorporated into the captioning model, one at a time, and then tested.

3.1 The Adopted Network Architectures

In this work, we use the top three pre-trained CNNs in the Keras library[†] in both Top-1 and Top-5 accuracy (NASNetLarge, InceptionResNet V2 and Xception) and add Inception V3. Also, from the results of [15], we include the top two models from the VGG Net architecture (VGG16 and VGG19), all the models with which they experimented from the ResNet architecture (ResNet50, ResNet101 and ResNet152) and the top three models from the DenseNet architecture (DenseNet121, DenseNet169 and DenseNet201). The total is 12 models, but only one is in use at a time.

VGG [25] is one of the classical CNN architectures, known for its simplicity. The network consists of small 3×3 filters, pooling layers and a fully connected layer. VGG16 has 16 layers, while VGG19 has 19 layers.

ResNets [26] (short for Residual Networks) use residual connections, which sum the output of a block of layers with its input and pass it as input to the subsequent layer. ResNets learn a residual mapping instead of hoping that a group of stacked layers directly fits a desired underlying mapping. The number of the model (50, 101 and 152) refers to the number of layers in the model.

Inception V3 [27], from the Inception family, is a convolutional neural network for image classification. It has 48 layers and uses symmetric and asymmetric blocks that include convolutional layers, average pooling layers, max pooling layers, concatenation, dropouts and fully connected layers at the end.

Xception [28] (Extreme version of Inception) has 71 layers with a modified depth-wise separable convolution method instead of Inception modules. It takes inspiration from Inception V3 and outperforms it through better use of model parameters.

The InceptionResNet model [29] combines the Inception structure with the residual connection method. In InceptionResNet, Residual connections are combined with convolutional filters of multiple sizes to form the Inception-ResNet block. Residual connections reduce the time of training and avoid the degradation problem

[†] Available at `https://keras.io/`

that deep structures produce.

The DenseNet architecture [30] is a convolutional neural network that employs dense connections between layers using “Dense Blocks”, in which all layers are connected with each other directly. Every layer takes additional inputs from all layers before it and gives input to all succeeding layers. The numbers 121, 169 and 201 denote the depth of the model.

NASNetLarge [31] uses the technique of Neural Architecture Search (NAS), in which the blocks of the CNN are searched by reinforcement learning. It uses two types of convolutional cells to formulate feature maps: normal cells that return maps of the same height and width, and reduction cells, where the height and width of the feature map are reduced by a factor of two.

For the image captioning model, we focus on the methodology of “Show, attend and tell” [32] with some modification. We chose this model because it is simple, fast to train and evaluate, and uses attention to generate captions. So it can represent image captioning systems that adopt an Encoder-Decoder architecture with attention.

For the encoding part, the model employs a convolutional neural network for feature extraction, without fine-tuning. It produces a feature map from the last layer before the fully connected layer. The output of the feature extraction phase is L vectors, with each vector being a D-dimensional representation that corresponds to a part of the image. The model learns an embedding space of length 256 using one fully connected layer.

For the decoding part, we use a Gated Recurrent Unit (GRU) [33] instead of an LSTM in [32] to exploit the speed and low memory usage in a GRU. It produces a caption by generating one word at every time step, conditioned on a context vector, the previous hidden state, and the previously generated words. The model can be trained using the backpropagation algorithm deterministically.

For attention, we use the Bahdanau soft attention as introduced in [34]. It computes a soft attention weighted annotation vector using the formula:

$$\Phi(\{a_{(i)}\}, \{\alpha_{(i)}\}) = \sum_{i}^{L} \alpha_{(i)} a_{(i)} \tag{1}$$

This deterministic attention makes the model as a whole smooth and differentiable.

3.2 Datasets

The datasets that we used are three of the most used in image captioning: Flickr8k [6], Flickr30k [7] and MS COCO [8]. They are all collected from the Flickr photo sharing website and consist of real-life images, annotated by humans (five annotations per image). Table 1 contains a brief comparison. It is worth noting that MS COCO does not publish the labels of the testing set.

Dataset	Training split	Validation Split	Testing Split	Total Images
Flickr8k	7k	1k	1k	8k
Flickr30k	28k	1k	1k	30k
MS COCO	83k	41k	41k	144k

Table 1: A comparison of the used datasets.

3.3 Pre-processing

In this section, we present the performed pre-processing steps on the data in this work:

1. Randomly sort the dataset, in pairs of image-caption.
2. Decode the images.
3. Resize the images to the size that the CNN expects. Every CNN has its own expected size.
4. Tokenise the text. For each sentence in the text, it is split into tokens by punctuation, special characters and white space.
5. Count the tokens, sort them by frequency and select the 15,000 words with the highest frequency as the vocabulary of the system. This helps to eliminate words that are less likely to be needed and prevents over-fitting.
6. Generate word-to-index and index-to-word structures. After that, they are used to convert token sequences into word-id sequences.
7. Padding: sequences of identifiers are padded at the end, so that all sequences have the same length.

4 Experiments and Results

The experiments were run on a mainframe with 32 GB of memory, an Intel Core i9-9900K CPU and an NVIDIA GeForce RTX 2080 GPU. The code is written in the Python programming language and TensorFlow library. All pre-trained models were provided by the Keras library. We report the results on three benchmark datasets, using the evaluation and testing sets, on 12 pre-trained models, resulting in 72 experiments.

We used different evaluation metrics that are commonly used in the image captioning domain. BLEU metrics [35] are widely used in automatic evaluation of machine-translated text and measure the correspondence between a machine translation output and a human translation, in the case of image captioning the machine translation output corresponds to the automatically-generated caption and the human translation corresponds to the human description of the image. METEOR [36] is calculated using the harmonic mean of unigram precision and recall with a higher weight for the recall than that of the precision, it is calculated as follows:

$$METEOR = \frac{10 \times Precision \times Recall}{Recall + 9 \times Precision} \tag{2}$$

ROUGE-L [37] evaluates the adequacy and fluency of the generated text through a Longest Common Subsequence (LCS) score, whereas CIDEr [38] focuses on grammaticality and saliency. SPICE [39] analyses the semantics of the generated text through constructing a “scene graph” for both the original caption and the generated caption, and then matches the words only if their lemmatised WordNet representations are equal. BLEU, METEOR and ROUGE are not well correlated with human assessments of quality, whereas SPICE and CIDEr have better correlation, but tend to be harder to optimise. We multiplied the scores by 1000 to avoid redundancy.

The model was trained and tested for each combination of datasets and CNNs, once trained on the training set and then tested on the validation set, and once trained on the training and validation sets combined and then tested on the testing set.

We report three tables describing the results on Flickr8k, Flickr30k and MS COCO. The best three results of each evaluation metric on the testing sets are written in boldface. The discussion will consider the testing results, but the development set scores are also reported to reflect the model’s ability to generalise. The testing is done after training on the training and development sets combined. As the MS COCO dataset does not provide labels for testing data, the testing set used here is the validation set (about 41,000 images), unlike many previous works that used Karpathy's split [40] (1000 testing images) because of the importance of large testing data in our experiment. The features were extracted from the last layer after removing the fully connected layer, and the captioning model was trained on a vocabulary size of 15,000 words.

Table 2 presents the results of the different architectures on the Flickr8k dataset. We sorted the scores and presented the results in Figure 2. It is notable that ResNet101, ResNet152, DenseNet201 and Xception take the lead in all of the metrics.

We use the following abbreviations in figures: RN50: ResNet50, RN101: ResNet101, RN152: ResNet152, IncV3: Inception V3, Xcep: Xception, IncRNV2: InceptionResNet V2, DN121: DenseNet121, DN169: DenseNet169, DN201: DenseNet201, NASNL: NASNetLarge.

Model (Train/Test)	**BLEU-1**		**BLEU-2**		**BLEU-3**		**BLEU-4**		**METEOR**		**CIDEr**		**ROUGE-L**		**SPICE**	
VGG16	607	393	459	219	336	112	239	55	238	135	671	168	480	311	177	82
VGG19	576	416	434	239	316	131	225	71	244	139	627	178	477	316	186	88
ResNet50	638	404	503	232	387	127	293	68	264	143	797	198	519	322	203	94
ResNet101	654	**449**	516	**265**	399	**150**	304	**79**	259	**153**	796	**231**	519	**346**	200	**99**
ResNet152	653	**445**	517	**263**	399	**148**	301	**79**	264	**155**	815	**249**	521	**349**	204	**103**
Inception V3	604	411	457	235	336	129	243	70	243	139	683	198	484	319	183	88
Xception	631	440	492	259	373	**147**	278	**80**	252	**151**	742	224	501	**341**	196	97
InceptionResNet V2	541	434	397	250	285	135	202	72	231	145	575	219	451	331	176	95
DenseNet121	609	410	466	230	349	123	257	63	244	144	714	204	489	320	184	94
DenseNet169	593	426	453	250	337	139	245	74	250	148	709	209	491	337	190	95
DenseNet201	634	**447**	492	**261**	373	144	278	72	249	150	759	**230**	506	340	190	96
NASNetLarge	698	390	578	219	472	119	379	62	290	148	984	191	568	314	222	**99**

Table 2: Experimental results on the Flickr8k dataset.

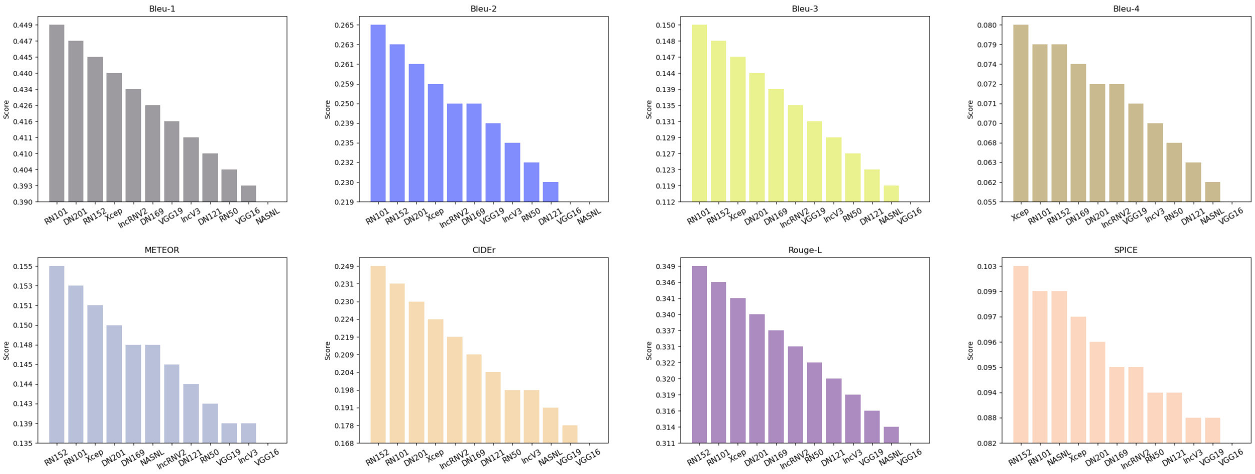


Figure 2: Sorted scores on the Flickr8k dataset.

In Table 3, we present the scores of the different architectures on the Flickr30k dataset, using the studied evaluation metrics. Figure 3 shows the sorted scores. We can see that Xception, NASNetLarge and Inception-ResNet V2 take many of the first three places.

Model (Train/Test)	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		CIDEr		ROUGE-L		SPICE	
VGG16	457	360	287	183	179	93	110	47	155	116	279	102	342	270	98	62
VGG19	456	379	284	197	175	98	107	51	155	119	272	106	338	279	98	63
ResNet50	513	380	343	208	228	**111**	150	**58**	174	**130**	366	133	379	**298**	117	**72**
ResNet101	494	376	329	206	216	109	141	**58**	180	125	353	122	380	288	121	71
ResNet152	497	376	332	199	220	103	144	52	183	121	358	125	385	282	123	67
Inception V3	479	390	308	205	196	104	124	52	167	123	307	127	364	287	111	71
Xception	485	**399**	319	**220**	208	**117**	135	**62**	176	123	337	**148**	372	293	118	**74**
InceptionResNet V2	474	**395**	297	**213**	185	**111**	114	57	158	**131**	287	**150**	348	**294**	104	**76**
DenseNet121	445	374	281	203	176	107	109	55	165	125	269	120	349	289	107	71
DenseNet169	469	392	300	211	188	108	117	56	165	125	302	129	354	**295**	109	68
DenseNet201	477	384	305	205	192	106	120	55	164	122	311	134	355	289	107	68
NASNetLarge	515	**397**	349	**216**	237	**114**	160	**60**	187	**126**	402	**160**	398	293	129	71

Table 3: Experimental results on the Flickr30k dataset.

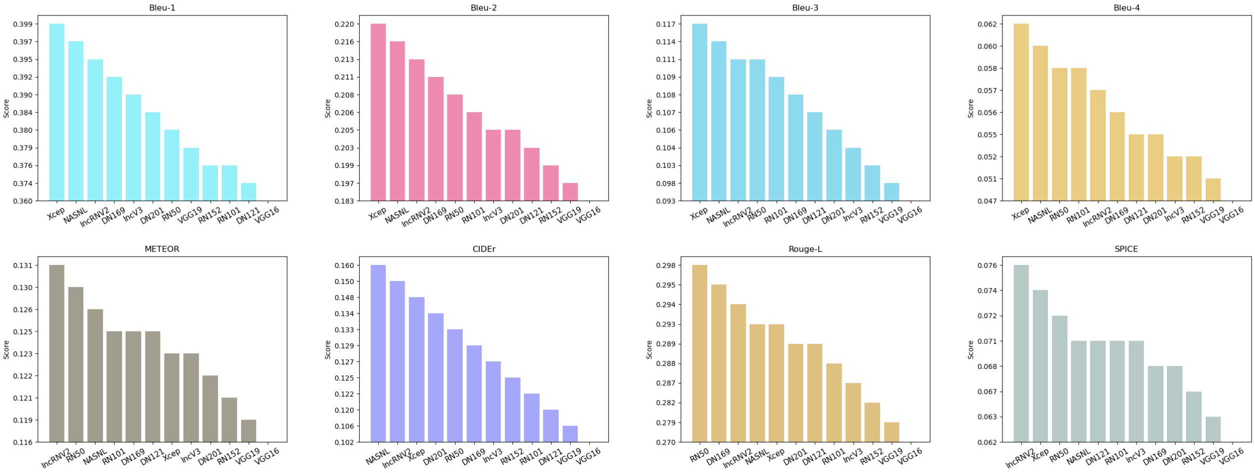


Figure 3: Sorted scores on the Flickr30k dataset.

Table 4 presents the results of the different architectures on the MS COCO dataset. In Figure 4, we can notice that NASNetLarge performs the best in all metrics and InceptionResNet V2 consistently scores among the highest in all of the metrics as well. For the BLEU scores, InceptionResNet V2 and Xception score the second and third place, respectively. Because MS COCO is the largest among the datasets that we used, we consider that scores on it have higher reliability than the results on Flickr8k and Flickr30k.

Model (Train/Test)	**BLEU-1**		**BLEU-2**		**BLEU-3**		**BLEU-4**		**METEOR**		**CIDEr**		**ROUGE-L**		**SPICE**	
VGG16	499	447	318	263	196	150	119	84	169	149	391	282	367	332	112	94
VGG19	502	449	321	266	199	152	121	85	174	152	409	293	374	338	117	96
ResNet50	534	476	351	282	224	162	141	92	194	159	508	343	404	352	134	102
ResNet101	525	474	344	288	219	168	137	96	190	167	488	351	396	**362**	132	**108**
ResNet152	537	478	353	289	225	168	142	96	190	163	506	346	399	**359**	133	105
Inception V3	518	478	334	288	210	167	130	95	186	160	473	341	392	351	127	105
Xception	530	**484**	349	**291**	223	**170**	141	**98**	189	163	510	**362**	398	359	133	107
InceptionResNet V2	519	**485**	338	**295**	211	**173**	129	**99**	181	**164**	449	**358**	388	358	124	**109**
DenseNet121	518	476	333	287	207	167	127	96	181	**164**	450	346	384	358	124	106
DenseNet169	507	474	327	285	206	165	127	94	186	158	449	335	387	348	128	103
DenseNet201	520	474	336	288	212	168	132	97	178	163	453	341	384	356	122	106
NASNetLarge	572	**488**	394	**298**	265	**176**	177	**103**	205	**174**	605	**390**	428	**368**	147	**117**

Table 4: Experimental results on the MS COCO dataset.

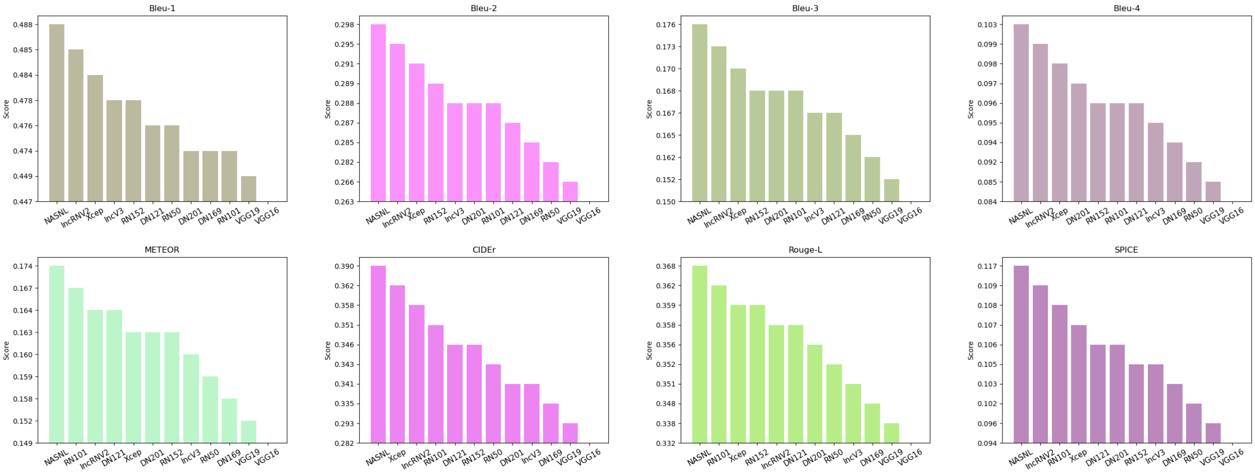


Figure 4: Sorted scores on the MS COCO dataset.

If we analyse the results in regard to each individual metric, we cannot conclude one best feature extraction model to optimise all metrics, so we prefer models that keep a high ranking across different datasets. We focus in our analysis on the "reliability" of the feature extraction model across datasets.

BLEU-1. For Flickr8k, ResNet101, DenseNet201 and ResNet152 score the highest, while scoring among the lowest in Flickr30k and MS COCO. NASNetLarge scores very high in Flickr30k and MS COCO but the lowest in Flickr8k. However, Xception and InceptionResNet V2 retain their positions in the top half.

BLEU-2. Xception and InceptionResNet V2 retain places in the top half. As in the case of BLEU-1, NASNetLarge does unreliably appear on the top.

BLEU-3. For BLEU-3, the unreliable top status of NASNetLarge and ResNet152 is also apparent. ResNet101, Xception and InceptionResNet V2 are always in the top half.

BLEU-4. Xception takes the top place in Flickr8k and Flickr30k and the fourth rank in MS COCO. Xception, ResNet101 and InceptionResNet V2 stay in the top half.

We can notice that Xception scores well across all BLEU metrics. In general, models that score well on one BLEU metric performed well on the other BLEU metrics. This is an expected consequence of the relatedness of the BLEU metrics.

METEOR. ResNet101 and NASNetLarge are the only ones always in the top half, while other models greatly change ranks.

CIDEr. For CIDEr, only Xception and InceptionResNet V2 retain places in the top half.

ROUGE-L. In the case of ROUGE-L, only Xception and InceptionResNet V2 remain in the top half. The others vary greatly in rankings.

SPICE. ResNet101, NASNetLarge and Xception remain reliably in the top half.

Table 5 contains a summary of our recommended models for optimising each evaluation metric. For each metric, we list the models that always appeared in the top 6 ratings across the three different datasets. We use this simple heuristic of recommending a specific model because of the lack of a unified evaluation metric for image captioning, and the diversity between the results across datasets. In these results, Xception, followed by InceptionResNet V2, appear to yield the most robust features for image captioning.

Metric	**Recommended Models**
BLEU-1	Xception, InceptionResNet V2
BLEU-2	Xception, InceptionResNet V2
BLEU-3	ResNet101, Xception, InceptionResNet V2
BLEU-4	ResNet101, Xception, InceptionResNet V2
METEOR	ResNet101, NASNetLarge
CIDEr	Xception, InceptionResNet V2
ROUGE-L	Xception, InceptionResNet V2
SPICE	ResNet101, NASNetLarge, Xception

Table 5: A summary of the recommended models for optimising each evaluation metric.

Interestingly, the results that we find in our work are consistent with the results of [24], in which Ke et al. found a strong influence of the model family on the results, more than the size of the model. They demonstrated that architectures that work for ImageNet do not necessarily work for medical imaging tasks, which resembles the superiority of certain architectures on each dataset in our experiment results. They also reported that newer architectures generated from NAS on ImageNet (EfficientNet, MobileNet and MNASNet) underperformed DenseNets and ResNets, which is consistent with our result that NASNetLarge performed the best in MS COCO, which is the largest of the used datasets in our work.

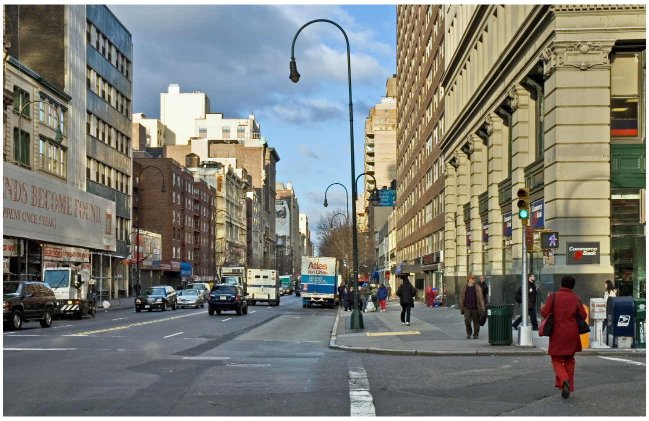
Unlike the work of Holliday and Dudek [15] and the work of Valev et al. [13], we do not recommend DenseNet121 nor DenseNet161 as feature extractors because of their relatively low performance. Although DenseNet161 was not in our experiment set, but none of the DenseNet variations performed consistently well across the datasets in this experiment to be in our recommendation list. Irvin et al. [18] also recommended DenseNet121. We justify the disagreement between their recommendations and ours by the difference in the domain of use. In [15], Holliday and Dudek measured robustness for a) Scale and perspective, and b) Appearance. In [13], Valev et al. worked with fine-grained vehicle classification, while in [18], Irvin et al. worked on the detection of the presence of 14 observations in medical chest images. So, none of them tried to actually generate descriptive text from an image.

The results of Sharif et al. [21] on Flickr30k conform to our results, in which they found that NASNetLarge performed the best on BLEU-1, ROUGE-L, METEOR and CIDEr. In order to give a comparison of the captioning output using the different feature extraction approaches, we applied them on a new example of an every-day image randomly selected from Wikipedia.[‡] Figure 5 presents the input image along with a table giving one caption output for each feature extraction CNN approach. Among the captions presented, the Xception and

[‡] Image source: `https://upload.wikimedia.org/wikipedia/commons/e/e1/NYC_14th_Street_looking_west_12_2005.jpg`

InceptionResNet V2 captions seem the most accurate.

It can be noticed that VGG, ResNet and DenseNet169 captions have grammatical errors. ResNet152 and Inception V3 captions have some repetition. DenseNet201 captured the nature of the image (“outdoor”, “blue shirt”, “car”, “way”, “posing”) correctly but failed to describe it. NASNetLarge captured that there are people in the picture.



Model	Caption
VGG16	with carefully rubber garb side suit dog shirt smiling streamers while clothing offering about $< end >$.
VGG19	a sizes in water products on water back sitting on crawling written crawling on into a picture and some sort into into at at that lounge on over into be on water products at that photo surfing something products on and water pans take products under $< end >$.
ResNet50	a open at a while talk on a while playing for hand out his head of an garb picking with a luggage moving two jean while at a corner of a chair at his board front of an knees with a plate $< end >$.
ResNet101	some at outside a farm surrounded by a child sits to a cute sitting on next to a adult and eating of a person is standing on to a cellphone is reaching with a standing at a on and the is eating something like a white behind a cell is over her.
ResNet152	are standing and hiding a meal holding is standing near a chair at the floor next to a next to a chair giving is in an old clothes in to a giving a surfboard giving a animals sitting in to a scale before a jacket man giving a scale while an ducks.
Inception V3	a frisbee in the camera in the camera in front of a feeding a fire in front of a frisbee in the camera with a lone frisbee in an open to climb inside of a frisbee in front of a tennis out to the at the camera and another person standing outside.
Xception	a girl holding a large and another walks off on the front $< end >$.
InceptionResNet V2	a person posing in some holding a bright orange flowers $< end >$.
DenseNet121	a plane while standing in the cool shirt is on the top of a rope from an wild bends at a horse and flowers in front of a skateboard on a chair looking up to see another man is standing near a pair of a blue outfit is post from a gray.
DenseNet169	large pile of a small dark room in the edge of a white bowl of a deck of a wooden railing of a cake there outfit next to get his phone in front of a kitchen next to be a deck near a vase next to touch if the $< unk >$ of a.
DenseNet201	an outdoor tennis on a high side of a large cell phone are playing a chair with a wooden bench with a beautiful blue shirt is ride on a motor car on the edge of a new coca cola way that is posing in a hand $< end >$.
NASNetLarge	some kind of the man in the next to get up on a cowboy hat holds some people are on a hose sitting in the man in the milk on a tripod taking a man sits beside a long legs on stands in a hose to be getting gear drinking from a.

Figure 5: An example image with its captions for each feature extraction CNN. The "$< unk >$" token means a word our of the vocabulary.

5 Conclusions

For comparing the convolutional neural networks as feature extractors for the image captioning task, we performed 72 experiments on 12 CNNs pre-trained on the ImageNet dataset. The features were extracted from the last layer after removing the last fully connected layer. The image captioning model was trained and tested on three datasets and evaluated with eight different evaluation metrics that capture different aspects of image captioning quality. The results indicate a very strong relationship between the nature of the data and the structure of the feature extraction model in use. When possible, an image captioning model can be optimised for the data by using the CNN adequate for the dataset in question. However, in this work there are CNNs that have a certain degree of reliability for optimising a specific metric regardless of the dataset in question. On the MS COCO dataset, which is the largest in the experiment, NASNetLarge performed the best in all of the metrics. Xception and InceptionResNet V2 gave the most robust features across datasets. They performed consistently well in most of the metrics for the three datasets.

Future work can explore what makes certain CNN models perform better on certain datasets than others, and develop a more advanced CNN architecture that produces better and more robust features. We also recommend running experimental studies on the same datasets for other computer vision tasks such as object/event recognition or 3D scene reconstruction from real-world images.

References

[1] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999. doi:10.1109/ICCV.1999.790410.

[2] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *Computer Vision – ECCV 2006*, pp. 404–417, Springer Berlin Heidelberg, 2006. doi:10.1007/11744023_32.

[3] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, “Deep convolutional ranking for multilabel image annotation,” *arXiv preprint arXiv:1312.4894*, 2013. doi:10.48550/arXiv.1312.4894.

[4] A. Shin, Y. Ushiku, and T. Harada, “Image captioning with sentiment terms via weakly-supervised sentiment dataset.,” in *BMVC*, 2016. doi:10.5244/C.30.53.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi:10.1109/CVPR.2009.5206848.

[6] M. Hodosh, P. Young, and J. Hockenmaier, “Framing image description as a ranking task: Data, models and evaluation metrics,” *Journal of Artificial Intelligence Research*, vol. 47, pp. 853–899, 2013. doi:10.1613/jair.3994.

[7] B. A. Plummer, L. Wang, C. M. Cervantes, J. C. Caicedo, J. Hockenmaier, and S. Lazebnik, “Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2641–2649, 2015. doi:10.1109/ICCV.2015.303.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *Computer Vision – ECCV 2014*, pp. 740–755, Springer International Publishing, 2014. doi:10.1007/978-3-319-10602-1_48.

[9] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, “On the performance of convnet features for place recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4297–4304, 2015. doi:10.1109/IROS.2015.7353986.

[10] A. ”Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, “Neural codes for image retrieval,” in *Computer Vision – ECCV 2014*, pp. 584–599, Springer International Publishing, 2014. doi:10.1007/978-3-319-10590-1_38.

[11] Z. Chen, O. Lam, A. Jacobson, and M. Milford, “Convolutional neural network-based place recognition,” *Proceedings of the 16th Australasian Conference on Robotics and Automation 2014*, pp. 1–8, 2014.

[12] K. Tran, X. He, L. Zhang, and J. Sun, “Rich image captioning in the wild,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 434–441, 2016. doi:10.1109/CVPRW.2016.61.

[13] K. Valev, A. Schumann, L. Sommer, and J. Beyerer, “A systematic evaluation of recent deep learning architectures for fine-grained vehicle classification,” *Pattern Recognition and Tracking XXIX*, vol. 10649, pp. 1 – 11, 2018. doi:10.1117/12.2305062.

[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, “3d object representations for fine-grained categorization,” in *2013 IEEE International Conference on Computer Vision Workshops*, pp. 554–561, 2013. doi:10.1109/ICCVW.2013.77.

[15] A. Holliday and G. Dudek, “Pre-trained cnns as visual feature extractors: A broad evaluation,” in *2020 17th Conference on Computer and Robot Vision (CRV)*, pp. 78–84, 2020. doi:10.1109/CRV50864.2020.00019.

[16] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, PMLR, 09–15 Jun 2019.

[17] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, “Self-training with noisy student improves imagenet classification,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695, 2020. doi:10.1109/CVPR42600.2020.01070.

[18] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S. Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N. Patel, M. P. Lungren, and A. Y. Ng, “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 590–597, Jul. 2019. doi:10.1609/aaai.v33i01.3301590.

[19] P. Rajpurkar, A. Joshi, A. Pareek, P. Chen, A. Kiani, J. Irvin, A. Y. Ng, and M. P. Lungren, “Chexpedition: Investigating generalization challenges for translation of chest x-ray algorithms to the clinical setting,” *arXiv e-prints*, pp. arXiv–2002, 2020. doi:10.48550/arXiv.2002.11379.

[20] Y. Yi, H. Deng, and J. Hu, “Improving image captioning evaluation by considering inter references variance,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 985–994, Association for Computational Linguistics, jul 2020. doi:10.18653/v1/2020.acl-main.93.

[21] N. Sharif, M. A. A. K. Jalwana, M. Bennamoun, W. Liu, and S. A. A. Shah, “Leveraging linguistically-aware object relations and nasnet for image captioning,” in *2020 35th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, 2020. doi:10.1109/IVCNZ51579.2020.9290719.

[22] J. Zhang, K. Li, and Z. Wang, "Parallel-fusion lstm with synchronous semantic and visual information for image captioning," *Journal of Visual Communication and Image Representation*, vol. 75, p. 103044, 2021. doi:10.1016/j.jvcir.2021.103044.

[23] Y. Zhang, X. Shi, S. Mi, and X. Yang, "Image captioning with transformer and knowledge graph," *Pattern Recognition Letters*, vol. 143, pp. 43–49, 2021. doi:10.1016/j.patrec.2020.12.020.

[24] A. Ke, W. Ellsworth, O. Banerjee, A. Y. Ng, and P. Rajpurkar, "Chextransfer: Performance and parameter efficiency of imagenet models for chest x-ray interpretation," in *Proceedings of the Conference on Health, Inference, and Learning*, CHIL '21, p. 116–124, Association for Computing Machinery, 2021. doi:10.1145/3450439.3451867.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. doi:10.1109/CVPR.2016.90.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2016. doi:10.1109/CVPR.2016.308.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1800–1807, 2017. doi:10.1109/CVPR.2017.195.

[29] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, p. 4278–4284, AAAI Press, 2017. doi:10.5555/3298023.3298188.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, 2017. doi:10.1109/CVPR.2017.243.

[31] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8697–8710, 2018. doi:10.1109/CVPR.2018.00907.

[32] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, PMLR, 2015.

[33] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, oct 2014. doi:10.3115/v1/d14-1179.

[34] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[35] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Association for Computational Linguistics, jul 2002. doi:10.3115/1073083.1073135.

[36] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72, Association for Computational Linguistics, jun 2005.

[37] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, jul 2004.

[38] R. Vedantam, C. L. Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4566–4575, 2015. doi:10.1109/CVPR.2015.7299087.

[39] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Computer Vision – ECCV 2016*, (Cham), pp. 382–398, Springer International Publishing, 2016. doi:10.1007/978-3-319-46454-1_24.

[40] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 664–676, 2017. doi:10.1109/TPAMI.2016.2598339.