JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY

Image Captioning using Deep Learning

Project Report

Submitted By:

Vanshika Mittal 16103012

Shubham Kumar Mishra 16103220

Abhishek Verma 16103237

Aman Goyal 16103279

Submitted To:

Dr. Anuja Arora

Batch: B5-6

Problem Statement

In this project, we elaborate on image captioning concerning especially dense image captioning. Caption generation is a challenging artificial intelligence problem where a textual description must be generated for a given photograph.

It requires both methods from computer vision to understand the content of the image and a language model from the field of natural language processing to turn the understanding of the image into words in the right order. Recently, deep learning methods have achieved state-of-the-art results on examples of this problem.

We present technical fundamentals of a model striving to solve such a task. Concretely, a detailed structure of Neural Image Caption is discussed. Experimentally, we examine results of neural networks and analyse the model's weaknesses. We show that 92% of the generated captions are identical to a caption in the training set while the quality of those and the novel ones remains the same.

Keywords: Image captioning, dense captioning, convolutional neural networks, long short-term memory.

Research Papers

Title: Show and Tell: A Neural Image Caption Generator

Authors: Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan

Datasets: For evaluation we use a number of datasets which consist of images and sentences in English describing these images.

The statistics of the datasets are as follows:

Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [6]	-	-	1000
Flickr8k [26]	6000	1000	1000
Flickr30k [33]	28000	1000	1000
MSCOCO [20]	82783	40504	40775
SBU [24]	1M	-	-

With the exception of SBU, each image has been annotated by labellers with 5 sentences that are relatively visual and unbiased. SBU consists of descriptions given by image owners when they uploaded them to Flickr. As such they are not guaranteed to be visual or unbiased and thus this dataset has more noise.

The Pascal dataset is customary used for testing only after a system has been trained on different data such as any of the other four dataset. In the case of SBU, we hold out 1000 images for testing and train on the rest. Similarly, we reserve 4K random images from the MSCOCO validation set as test, called COCO-4k, and use it to report results in the following section.

Methodology: NIC (Neural Image Captioner) is based on a convolution neural network that encodes an image into a compact representation, followed by a recurrent neural net- work that generates a corresponding sentence. The model is trained to maximize the likelihood of the sentence given the image. Experiments on several datasets show the robust- ness of NIC in terms of qualitative results (the generated sentences are very reasonable) and quantitative evaluations, using either ranking metrics or BLEU, a metric used in ma- chine translation to evaluate the quality of generated sentences.

Results: Since our model is data driven and trained end-to-end, and given the abundance of datasets, we wanted to answer questions such as “how dataset size affects generalization”, “what kinds of transfer learning it would be able to achieve”, and “how it would deal with weakly labeled examples”. As a result, we performed

experiments on five different datasets, which enabled us to understand our model in depth.

Title: Where to put the Image in an Image Caption Generator

Authors: Marc Tanti, Albert Gatt, Kenneth P. Camilleri

Datasets: The datasets used for all experiments were the version of Flickr8K (Hodosh et al., 2013), Flickr30K (Young et al., 2014), and MSCOCO (Lin et al., 2014) distributed by Karpathy and Fei-Fei (2015).6 All three datasets consist of images taken from Flickr combined with between five and seven manually written captions per image. The provided datasets are split into a training, validation, and test set using the following number of images respectively: Flickr8K - 6000, 1000, 1000; Flickr30K - 29000, 1014, 1000; MSCOCO - 82783, 5000, 5000. The images are already vectorised into 4096-element vectors via the activations of layer 'fc7' (the penultimate layer) of the VGG OxfordNet 19-layer convolutional neural network (Simonyan and Zisserman, 2014), which was trained for object recognition on the ImageNet dataset (Deng et al., 2009).

The known vocabulary consists of all the words in the captions of the training set that occur at least 5 times. This amounts to 2539 tokens for Flickr8K, 7415 tokens for Flickr30K, and 8792 tokens for MSCOCO. These words are used both as inputs, which are embedded and fed to the RNN, and as outputs, which are assigned probabilities by the softmax function. Any other word which is not part of the vocabulary is replaced with an UNKNOWN token.

Methodology: In this paper author empirically showed that it is not especially detrimental to performance whether one architecture is used or another. The merge architecture does have practical advantages, as conditioning by merging allows the RNN's hidden state vector to shrink in size by up to four times.

Results: Three runs of each experiment, on each of the three datasets, were performed. For the various evaluation measures, we report the mean together with the standard deviation (reported in parentheses) over the three runs. For each run, the initial model weights, minibatch selections, and dropout selections are different since these are randomly determined. Everything else is identical across runs.

Title: Learning CNN-LSTM Architectures for Image Caption Generation

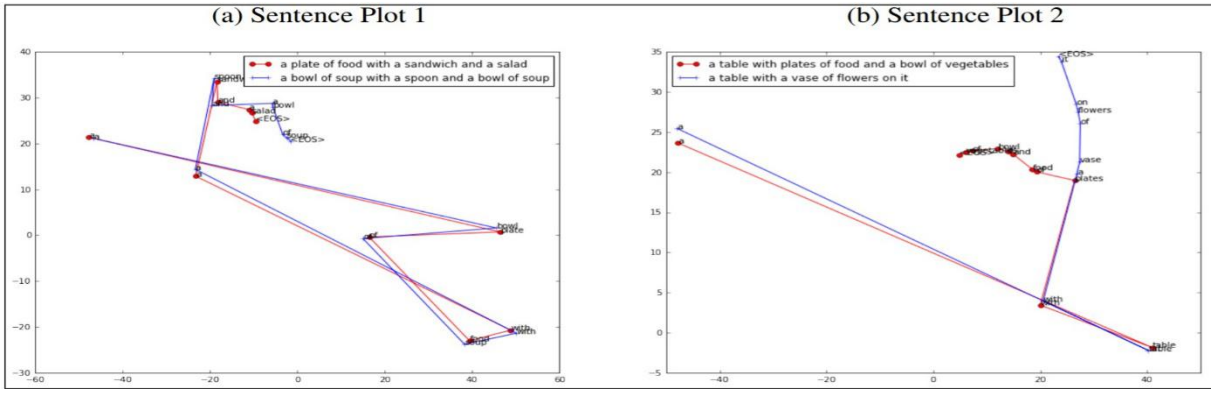
Author: Moses Soh

Dataset: We measure the performance of this architecture on the Microsoft Common Objects in Context (MSCOCO) dataset. The MSCOCO data comprises 82783 training images, 40504 validation images and 40775 test images. Each image is accompanied by at least five captions of varying length. In the training set, there are 414113 captions in total, for an average of 5.002 captions per image. We preprocess the caption dataset by replacing words that appear less than five times in the training

dataset with an ¡UNK¿ token, prepend each sentence with a SOS token, and append each sentence with a EOS token. The final vocabulary size is 8843. The mean and median length of the post-processed captions is 12.55 and 12 respectively. Figure 2 plots the histogram of caption lengths. There are long right tails in the empirical distribution, with the maximum caption length topping out at 57. We train all models on the entire training dataset and tune our hyperparameters on the validation set. We hold out 4000 images from the validation set as our test set.

Methodology: We use our best model to generate captions from the validation set on unseen images. In the caption generation process, we feed an image into our CNN which produces an encoding that is fed into our LSTM for decoding. As words are emitted, we feed the current predicted sentence back into the LSTM for it to predict the next word. As we feed each new predicted word into the LSTM, we record the hidden state that results from the combination of the new word and the previous hidden state. We do this for the 4000 held-out images in our validation set. These have a 7 median sentence length of 12 leading to approximately 48000 hidden states of dimension 512 × 1. We use t-SNE [18] on the entire 48000 × 512 matrix to reduce the space of hidden states to two dimensions, and then generate plots of the hidden state for each sentence.

Result:



The charts in the middle column of Figure 5 contain the projection of the hidden states of a sentence into two dimensions. The first sentence pair demonstrates that emitted words with close semantic meanings (’plate’ and ’bowl’, ’food’ and ’soup’) move the hidden state in the same direction, despite being conditioned on different image vectors. The second sentence pair shows that the emitted sequence ’a table with’ moves the hidden dimension in similar ways and that the hidden state sentence representation only diverges once the sentence begins to describe differences in the images. For example, the two line graphs diverge substantially only once the unrelated concepts of ’food’ and ’vase’ are emitted.

Title: Neural Image Caption Generation with Weighted Training and Reference

Author: Guiguang DingMinghai ChenSicheng ZhaoHui ChenJungong HanQiang Liu

Datasets: MS COCO and Flickr30k

Flickr30k dataset contains 31,783 images, while the more challenging MS COCO dataset consists of 123,287 images. Each image is labeled with at least five captions by different Amazon Mechanical Turk (AMT) workers. Since there is no standardized split on both datasets, we follow the publicly available split2 as in on Flickr30k dataset and in on MS COCO for fair comparison. That is, 1000 images are used for validation, 1000 for testing and the rest for training in Flickr30; 5000 images are selected for validation, 5000 for testing, and the rest for training in MS COCO.

Methodology:

Existing CNN-RNN framework-based methods suffer from two main problems: in the training phase, all the words of captions are treated equally without considering the importance of different words; in the caption generation phase, the semantic objects or scenes might be misrecognized. In our paper, we propose a method based on the encoder-decoder framework, named Reference based Long Short Term Memory (R-LSTM), aiming to lead the model to generate a more descriptive sentence for the given image by introducing reference information. Specifically, we assign different weights to the words according to the correlation between words and images during the training phase. We additionally maximize the consensus score between the captions generated by the captioning model and the reference information from the neighbouring images of the target image, which can reduce the misrecognition problem.

Results:

The results show that the proposed approach can outperform the state-of-the-art approaches on all metrics, especially achieving a 10.37% improvement in terms of CIDEr on MS COCO. By analyzing the quality of the generated captions, we come to a conclusion that through the introduction of reference information, our model can learn the key information of images and generate more trivial and relevant words for images

---

Title: Automatic Image Captioning

Author: Jia-Yu Pant, Hyung-Jeong Yang', Pinar Duygulu' and Christos Faloutsost

Datasets: We learn the association between image regions and words from manuallv annotated images.

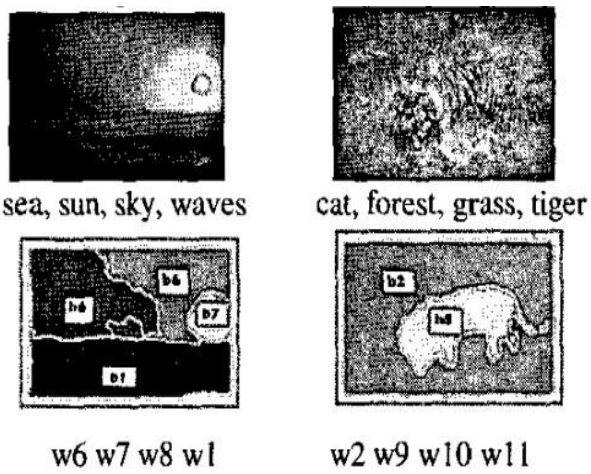


Figure 1. *Top: annotated images with their captions, bottom: corresponding blob-tokens and word tokens.*

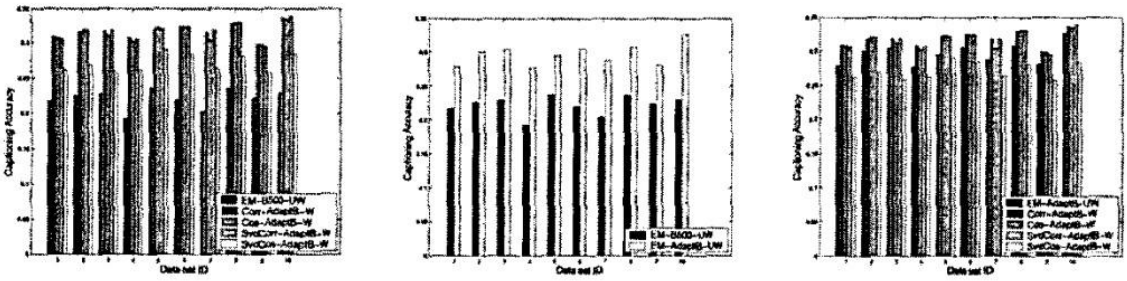
An image region is represented by a vector of features regarding its color, texture, shape, size and position. These feature vectors are clustered into B clusters and each region is assigned the label of the closest cluster center as in [3]. These labels are called blob-tokens. Formally, let 141 lr...rIN) be a set of annotated images where each image Ii is annotated with a set of terms Wi =(w,,~ ._.., w.~;) and a set of blob tokens Bi=(bi,,, ..., biflj]. where L; is the number of words, and Mi is the number of regions in image Ii. The goal is to construct a model that captures the association between terms and the blob-tokens, given Wi's and Bcs.

Methodology: We proposed 4 methods (Corr, Cos, SvdCorr, SvdCos) to estimate a W-by-B translation table T, whose (i,jj-element can be viewed as p(w;lbjj, the probability we caption the term wi, given we see a blob-token bj Definition 3 (Method Corrj Let T,,,,=DwTDB, The correlation-based translation table Tcom is defined by normalizing each column of T,m,o such that each column sum up to 1. Note that the (iJ-element of T,.,, can be viewed as an estimate ofp(w,lbjj. T,= measures the association between a term and a blob-token by the co- occurrence counts. Another possible measure could be to see how similar the overall occurrence pattem (over the training images) of a term and a blob-token is. Such occurrence pattems are in fact the columns of DW or De, and the similarity can be taken as the cosine value between pairs of column vectors. Definition 4 (Method Cos) Let the i-th column of the matrix Dw (De) bed, (d,;). Let cosjd be the cosine value of the angle column vectors dwi and dBj, and let TWJ be a W-by-B matrix whose (i,j)-element T,o(i,j)=cos;,j Normalize the columns of Tqo such that each column sums up to 1, and we get the cosinesimilarity translation table T,. Singular Value Decomposition (SVD) decomposes a given matrix XI-, into a product of three matrices U, A, VT. That is, X= UAVT, where U=[UI ,..., uJ, and V=[vl, ..., VJ with orthonormal columns ui's. vi's. A is a diagonal matrix, A=diag(ol ,..., G~~(~,,,,)), where ai > 0, for j 5 rank(X), aid, for j Previous works [14] show that by setting small ai's to

zero, yielding an optimal low rank representation a , SVD could be used to clean up noise and reveal rank(X). 1988 informative structure in a matrix X and achieve

better performance in information retrieval applications. We propose to use SVD to suppress the noise in the data matrix D before leaning the association. Following the general rule-of- thumb, we keep the first r ai's which preserve the 90% variance of the distribution, and set the others to zero. In the following, we denote the data matrix after SVD as Dsvd=[DwJrdlDBJvd]. Definition 5 (Merltod SvdCorr and SvdCos) Method SvdCorr and SvdCos generate the translation table Tcomsvd and Trmsvd following the procedureoutlined in Definition 3 and 4, respectively. The only difference is that instead of starting with the weighted data matrix D, here the matrix Dsvd is used. Algorithm 1 (Captioning) Given a translation table TIwSl (W total number of terms: B: total number of blob-tokens), and the number of captioning terms m for an image. An image I' with 1 blob-tokens B'= {Vi,..., b',), can be captioned by: First, form a query vector q=[q,, ..., qslT, where qi is the count of the blob-token bj in the set B'. Then, compute the term-likelihood vector p=Tq, where p+,, _... pwlT. pi is viewed as the predicted likelihood of the term w,. Finally, we caption the image I' with the m terms corresponding to the highest m pts in the p vector.

Result: The experiments are performed on 10 Core1 image data sets. Each data set contains about 5200 training images and 1750 testing images. The sets cover a variety of themes ranging from urban scenes to natural scenes, and from artificial objects like jetlplane to animals. Each image has in average 3 captioning terms and 9 blobs.



(a) (b) (c)

Figure 2. *Captioning accuracy improvement (a) proposed methods vs. the baseline* **EM-B500-UW**, *(b) adaptive blob-token generation on* **EM** *vs. the baseline, (c) proposed methods vs.* **EM** *when the adaptively generated blob-tokens are used.*

Figure 2(a) compares the proposed methods with the baseline algorithm 131 which is denoted as EM-B500- UW or simply EM (which means EM is applied to an unweighted matrix, denoted UW, in which the number of blob tokens is 500, denoted as BSOO).

Figure 2(b) shows that the adaptively generated blob-tokens improve the captioning accuracy of the EM algorithm. The improvement is around 7.5% in absolute accuracy (34.1% relative improvement) over the baseline method (whose accuracy is about 22%).

Title: Show and Tell: Lessons Learned from the 2015 MSCOCO Image Captioning Challenge .

Author: Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan

Datasets: For evaluation we use a number of datasets which consist of images and sentences in English describing these images. The statistics of the datasets are as follows: With the exception of SBU, each image has been annotated by labelers with 5 sentences that are relatively visual and unbiased. SBU consists of descriptions given by image owners when they uploaded them to Flickr. As such they are not guaranteed to be visual or unbiased and thus this dataset has more noise. The Pascal dataset is customary used for testing only after a system has been trained on different data such as any of the other four dataset. In the case of SBU, we hold out 1,000 images for testing and train on the rest as used by [14]. Similarly,

we reserve 4 K random images from the MSCOCO validation set as test, called COCO-4k, and use it to report results in the following section.

Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [2]	-	-	1,000
Flickr8k [42]	6,000	1,000	1,000
Flickr30k [43]	28,000	1,000	1,000
MSCOCO [44]	82,783	40,504	40,775
SBU [18]	1M	-	-

Methodology: When we first submitted our image captioning paper to CVPR 2015, we used the best convolutional neural network at the time, known as GoogleLeNet [48], which had 22 layers, and was the winner of the 2014 ImageNet competition. Later on, an even better approach was proposed in [24] and included a new method, called Batch Normalization, to better normalize each layer of a neural network with respect to the current batch of examples, so asto be more robust to nonlinearities.

The new approach got significant improvement on the ImageNet task (going from 6.67 percent down to 4.8 percent top-5 error) and the MSCOCO image captioning task, improving BLEU-4 by 2 points absolute.

Image Model Fine Tuning

In the original set of experiments, to avoid overfitting we initialized the image convolutional network with a pretrained model (we first used GoogleLeNet, then switched to the better Batch Normalization model), but then fixed its parameters and only trained the LSTM part of the model on the MS COCO training set.

Result: Since our model is data driven and trained end-to-end, and given the abundance of datasets, we wanted to answer questions such as “how dataset size affects generalization”, “what kinds of transfer learning it would be able to achieve”, and “how it would deal with weakly labelled examples”. As a result, we performed experiments on five different datasets, explained in Section 4.2, which enabled us to understand our model in depth.

Novel and how our implemented work will be different/ enhanced than existing work

One measure that can be used to evaluate the skill of the model are BLEU scores. For reference, below are some ball-park BLEU scores for skillful models when evaluated on the test dataset (taken from the 2017 paper "Where to put the Image in an Image Caption Generator"):

- BLEU-1: 0.401 to 0.578.
- BLEU-2: 0.176 to 0.390.
- BLEU-3: 0.099 to 0.260.
- BLEU-4: 0.059 to 0.170.

We have used Inception ResNet V2 and Fasttext word embeddings to improve the accuracy.

BLEU - 1 : 0.539671
BLEU - 2 : 0.293452
BLEU - 3 : 0.203698
BLEU - 4 : 0.096276
Validation

BLEU - 1 : 0.548384
BLEU - 2 : 0.306833
BLEU - 3 : 0.213625
BLEU - 4 : 0.103650
Test

BLEU - 1 : 0.547827
BLEU - 2 : 0.307524
BLEU - 3 : 0.215886
BLEU - 4 : 0.104762
Train

Data Gathering Phase

Flickr8k_Dataset.zip (1 Gigabyte) An archive of all photographs.
Flickr8k_text.zip (2.2 Megabytes) An archive of all text descriptions for photographs.

Flicker8k_Dataset: Contains 8092 photographs in JPEG format.
Flickr8k_text: Contains a number of files containing different sources of descriptions for the photographs.

The dataset has a pre-defined training dataset (6,000 images), development dataset (1,000 images), and test dataset (1,000 images).

Data Augmentation techniques

The model we will develop will generate a caption given a photo, and the caption will be generated one word at a time. The sequence of previously generated words will be provided as input. Therefore, we will need a '*first word*' to kick-off the generation process and a '*last word*' to signal the end of the caption.

We will use the strings '*startseq*' and '*endseq*' for this purpose. These tokens are added to the loaded descriptions as they are loaded. It is important to do this now before we encode the text so that the tokens are also encoded correctly.

As of now, no augmentation has been used for images in dataset.

Neural Network Model architecture

Inception ResNet V2

225214464/225209952 [==============================] - 4s 0us/step

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 299, 299, 3)	0	
conv2d_1 (Conv2D)	(None, 149, 149, 32)	864	input_1[0][0]
batch_normalization_1 (BatchNor	(None, 149, 149, 32)	96	conv2d_1[0][0]
activation_1 (Activation)	(None, 149, 149, 32)	0	batch_normalization_1[0][0]
conv2d_2 (Conv2D)	(None, 147, 147, 32)	9216	activation_1[0][0]
batch_normalization_2 (BatchNor	(None, 147, 147, 32)	96	conv2d_2[0][0]
activation_2 (Activation)	(None, 147, 147, 32)	0	batch_normalization_2[0][0]
conv2d_3 (Conv2D)	(None, 147, 147, 64)	18432	activation_2[0][0]
batch_normalization_3 (BatchNor	(None, 147, 147, 64)	192	conv2d_3[0][0]
activation_3 (Activation)	(None, 147, 147, 64)	0	batch_normalization_3[0][0]
max_pooling2d_1 (MaxPooling2D)	(None, 73, 73, 64)	0	activation_3[0][0]
conv2d_4 (Conv2D)	(None, 73, 73, 80)	5120	max_pooling2d_1[0][0]
batch_normalization_4 (BatchNor	(None, 73, 73, 80)	240	conv2d_4[0][0]
activation_4 (Activation)	(None, 73, 73, 80)	0	batch_normalization_4[0][0]
conv2d_5 (Conv2D)	(None, 71, 71, 192)	138240	activation_4[0][0]
batch_normalization_5 (BatchNor	(None, 71, 71, 192)	576	conv2d_5[0][0]
activation_5 (Activation)	(None, 71, 71, 192)	0	batch_normalization_5[0][0]
max_pooling2d_2 (MaxPooling2D)	(None, 35, 35, 192)	0	activation_5[0][0]
conv2d_9 (Conv2D)	(None, 35, 35, 64)	12288	max_pooling2d_2[0][0]
batch_normalization_9 (BatchNor	(None, 35, 35, 64)	192	conv2d_9[0][0]
activation_9 (Activation)	(None, 35, 35, 64)	0	batch_normalization_9[0][0]
conv2d_7 (Conv2D)	(None, 35, 35, 48)	9216	max_pooling2d_2[0][0]

conv2d_10 (Conv2D)	(None, 35, 35, 96)	55296	activation_9[0][0]
batch_normalization_7 (BatchNor	(None, 35, 35, 48)	144	conv2d_7[0][0]
batch_normalization_10 (BatchNo	(None, 35, 35, 96)	288	conv2d_10[0][0]
activation_7 (Activation)	(None, 35, 35, 48)	0	batch_normalization_7[0][0]
activation_10 (Activation)	(None, 35, 35, 96)	0	batch_normalization_10[0][0]
average_pooling2d_1 (AveragePoo	(None, 35, 35, 192)	0	max_pooling2d_2[0][0]
conv2d_6 (Conv2D)	(None, 35, 35, 96)	18432	max_pooling2d_2[0][0]
conv2d_8 (Conv2D)	(None, 35, 35, 64)	76800	activation_7[0][0]
conv2d_11 (Conv2D)	(None, 35, 35, 96)	82944	activation_10[0][0]
conv2d_12 (Conv2D)	(None, 35, 35, 64)	12288	average_pooling2d_1[0][0]
batch_normalization_6 (BatchNor	(None, 35, 35, 96)	288	conv2d_6[0][0]
batch_normalization_8 (BatchNor	(None, 35, 35, 64)	192	conv2d_8[0][0]
batch_normalization_11 (BatchNo	(None, 35, 35, 96)	288	conv2d_11[0][0]
batch_normalization_12 (BatchNo	(None, 35, 35, 64)	192	conv2d_12[0][0]
activation_6 (Activation)	(None, 35, 35, 96)	0	batch_normalization_6[0][0]
activation_8 (Activation)	(None, 35, 35, 64)	0	batch_normalization_8[0][0]
activation_11 (Activation)	(None, 35, 35, 96)	0	batch_normalization_11[0][0]
activation_12 (Activation)	(None, 35, 35, 64)	0	batch_normalization_12[0][0]
mixed_5b (Concatenate)	(None, 35, 35, 320)	0	activation_6[0][0] activation_8[0][0] activation_11[0][0] activation_12[0][0]
conv2d_16 (Conv2D)	(None, 35, 35, 32)	10240	mixed_5b[0][0]
batch_normalization_16 (BatchNo	(None, 35, 35, 32)	96	conv2d_16[0][0]
activation_16 (Activation)	(None, 35, 35, 32)	0	batch_normalization_16[0][0]
conv2d_14 (Conv2D)	(None, 35, 35, 32)	10240	mixed_5b[0][0]
conv2d_17 (Conv2D)	(None, 35, 35, 48)	13824	activation_16[0][0]
batch_normalization_14 (BatchNo	(None, 35, 35, 32)	96	conv2d_14[0][0]
batch_normalization_17 (BatchNo	(None, 35, 35, 48)	144	conv2d_17[0][0]
activation_14 (Activation)	(None, 35, 35, 32)	0	batch_normalization_14[0][0]
activation_17 (Activation)	(None, 35, 35, 48)	0	batch_normalization_17[0][0]
conv2d_13 (Conv2D)	(None, 35, 35, 32)	10240	mixed_5b[0][0]
conv2d_15 (Conv2D)	(None, 35, 35, 32)	9216	activation_14[0][0]
conv2d_18 (Conv2D)	(None, 35, 35, 64)	27648	activation_17[0][0]
batch_normalization_13 (BatchNo	(None, 35, 35, 32)	96	conv2d_13[0][0]
batch_normalization_15 (BatchNo	(None, 35, 35, 32)	96	conv2d_15[0][0]
batch_normalization_18 (BatchNo	(None, 35, 35, 64)	192	conv2d_18[0][0]
activation_13 (Activation)	(None, 35, 35, 32)	0	batch_normalization_13[0][0]
activation_15 (Activation)	(None, 35, 35, 32)	0	batch_normalization_15[0][0]
activation_18 (Activation)	(None, 35, 35, 64)	0	batch_normalization_18[0][0]
block35_1_mixed (Concatenate)	(None, 35, 35, 128)	0	activation_13[0][0] activation_15[0][0] activation_18[0][0]
block35_1_conv (Conv2D)	(None, 35, 35, 320)	41280	block35_1_mixed[0][0]
block35_1 (Lambda)	(None, 35, 35, 320)	0	mixed_5b[0][0]

			block35_1_conv[0][0]
block35_1_ac (Activation)	(None, 35, 35, 320)	0	block35_1[0][0]
conv2d_22 (Conv2D)	(None, 35, 35, 32)	10240	block35_1_ac[0][0]
batch_normalization_22 (BatchNo	(None, 35, 35, 32)	96	conv2d_22[0][0]
activation_22 (Activation)	(None, 35, 35, 32)	0	batch_normalization_22[0][0]
conv2d_20 (Conv2D)	(None, 35, 35, 32)	10240	block35_1_ac[0][0]
conv2d_23 (Conv2D)	(None, 35, 35, 48)	13824	activation_22[0][0]
batch_normalization_20 (BatchNo	(None, 35, 35, 32)	96	conv2d_20[0][0]
batch_normalization_23 (BatchNo	(None, 35, 35, 48)	144	conv2d_23[0][0]
activation_20 (Activation)	(None, 35, 35, 32)	0	batch_normalization_20[0][0]
activation_23 (Activation)	(None, 35, 35, 48)	0	batch_normalization_23[0][0]
conv2d_19 (Conv2D)	(None, 35, 35, 32)	10240	block35_1_ac[0][0]
conv2d_21 (Conv2D)	(None, 35, 35, 32)	9216	activation_20[0][0]
conv2d_24 (Conv2D)	(None, 35, 35, 64)	27648	activation_23[0][0]
batch_normalization_19 (BatchNo	(None, 35, 35, 32)	96	conv2d_19[0][0]
batch_normalization_21 (BatchNo	(None, 35, 35, 32)	96	conv2d_21[0][0]
batch_normalization_24 (BatchNo	(None, 35, 35, 64)	192	conv2d_24[0][0]
activation_19 (Activation)	(None, 35, 35, 32)	0	batch_normalization_19[0][0]
activation_21 (Activation)	(None, 35, 35, 32)	0	batch_normalization_21[0][0]
activation_24 (Activation)	(None, 35, 35, 64)	0	batch_normalization_24[0][0]
block35_2_mixed (Concatenate)	(None, 35, 35, 128)	0	activation_19[0][0] activation_21[0][0] activation_24[0][0]
block35_2_conv (Conv2D)	(None, 35, 35, 320)	41280	block35_2_mixed[0][0]
block35_2 (Lambda)	(None, 35, 35, 320)	0	block35_1_ac[0][0] block35_2_conv[0][0]
block35_2_ac (Activation)	(None, 35, 35, 320)	0	block35_2[0][0]
conv2d_28 (Conv2D)	(None, 35, 35, 32)	10240	block35_2_ac[0][0]
batch_normalization_28 (BatchNo	(None, 35, 35, 32)	96	conv2d_28[0][0]
activation_28 (Activation)	(None, 35, 35, 32)	0	batch_normalization_28[0][0]
conv2d_26 (Conv2D)	(None, 35, 35, 32)	10240	block35_2_ac[0][0]
conv2d_29 (Conv2D)	(None, 35, 35, 48)	13824	activation_28[0][0]
batch_normalization_26 (BatchNo	(None, 35, 35, 32)	96	conv2d_26[0][0]
batch_normalization_29 (BatchNo	(None, 35, 35, 48)	144	conv2d_29[0][0]
activation_26 (Activation)	(None, 35, 35, 32)	0	batch_normalization_26[0][0]
activation_29 (Activation)	(None, 35, 35, 48)	0	batch_normalization_29[0][0]
conv2d_25 (Conv2D)	(None, 35, 35, 32)	10240	block35_2_ac[0][0]
conv2d_27 (Conv2D)	(None, 35, 35, 32)	9216	activation_26[0][0]
conv2d_30 (Conv2D)	(None, 35, 35, 64)	27648	activation_29[0][0]
batch_normalization_25 (BatchNo	(None, 35, 35, 32)	96	conv2d_25[0][0]
batch_normalization_27 (BatchNo	(None, 35, 35, 32)	96	conv2d_27[0][0]
batch_normalization_30 (BatchNo	(None, 35, 35, 64)	192	conv2d_30[0][0]
activation_25 (Activation)	(None, 35, 35, 32)	0	batch_normalization_25[0][0]
activation_27 (Activation)	(None, 35, 35, 32)	0	batch_normalization_27[0][0]

activation_30 (Activation)	(None, 35, 35, 64)	0	batch_normalization_30[0][0]
block35_3_mixed (Concatenate)	(None, 35, 35, 128)	0	activation_25[0][0] activation_27[0][0] activation_30[0][0]
block35_3_conv (Conv2D)	(None, 35, 35, 320)	41280	block35_3_mixed[0][0]
block35_3 (Lambda)	(None, 35, 35, 320)	0	block35_2_ac[0][0] block35_3_conv[0][0]
block35_3_ac (Activation)	(None, 35, 35, 320)	0	block35_3[0][0]
conv2d_34 (Conv2D)	(None, 35, 35, 32)	10240	block35_3_ac[0][0]
batch_normalization_34 (BatchNo	(None, 35, 35, 32)	96	conv2d_34[0][0]
activation_34 (Activation)	(None, 35, 35, 32)	0	batch_normalization_34[0][0]
conv2d_32 (Conv2D)	(None, 35, 35, 32)	10240	block35_3_ac[0][0]
conv2d_35 (Conv2D)	(None, 35, 35, 48)	13824	activation_34[0][0]
batch_normalization_32 (BatchNo	(None, 35, 35, 32)	96	conv2d_32[0][0]
batch_normalization_35 (BatchNo	(None, 35, 35, 48)	144	conv2d_35[0][0]
activation_32 (Activation)	(None, 35, 35, 32)	0	batch_normalization_32[0][0]
activation_35 (Activation)	(None, 35, 35, 48)	0	batch_normalization_35[0][0]
conv2d_31 (Conv2D)	(None, 35, 35, 32)	10240	block35_3_ac[0][0]
conv2d_33 (Conv2D)	(None, 35, 35, 32)	9216	activation_32[0][0]
conv2d_36 (Conv2D)	(None, 35, 35, 64)	27648	activation_35[0][0]
batch_normalization_31 (BatchNo	(None, 35, 35, 32)	96	conv2d_31[0][0]
batch_normalization_33 (BatchNo	(None, 35, 35, 32)	96	conv2d_33[0][0]
batch_normalization_36 (BatchNo	(None, 35, 35, 64)	192	conv2d_36[0][0]
activation_31 (Activation)	(None, 35, 35, 32)	0	batch_normalization_31[0][0]
activation_33 (Activation)	(None, 35, 35, 32)	0	batch_normalization_33[0][0]
activation_36 (Activation)	(None, 35, 35, 64)	0	batch_normalization_36[0][0]
block35_4_mixed (Concatenate)	(None, 35, 35, 128)	0	activation_31[0][0] activation_33[0][0] activation_36[0][0]
block35_4_conv (Conv2D)	(None, 35, 35, 320)	41280	block35_4_mixed[0][0]
block35_4 (Lambda)	(None, 35, 35, 320)	0	block35_3_ac[0][0] block35_4_conv[0][0]
block35_4_ac (Activation)	(None, 35, 35, 320)	0	block35_4[0][0]
conv2d_40 (Conv2D)	(None, 35, 35, 32)	10240	block35_4_ac[0][0]
batch_normalization_40 (BatchNo	(None, 35, 35, 32)	96	conv2d_40[0][0]
activation_40 (Activation)	(None, 35, 35, 32)	0	batch_normalization_40[0][0]
conv2d_38 (Conv2D)	(None, 35, 35, 32)	10240	block35_4_ac[0][0]
conv2d_41 (Conv2D)	(None, 35, 35, 48)	13824	activation_40[0][0]
batch_normalization_38 (BatchNo	(None, 35, 35, 32)	96	conv2d_38[0][0]
batch_normalization_41 (BatchNo	(None, 35, 35, 48)	144	conv2d_41[0][0]
activation_38 (Activation)	(None, 35, 35, 32)	0	batch_normalization_38[0][0]
activation_41 (Activation)	(None, 35, 35, 48)	0	batch_normalization_41[0][0]
conv2d_37 (Conv2D)	(None, 35, 35, 32)	10240	block35_4_ac[0][0]
conv2d_39 (Conv2D)	(None, 35, 35, 32)	9216	activation_38[0][0]
conv2d_42 (Conv2D)	(None, 35, 35, 64)	27648	activation_41[0][0]

batch_normalization_37 (BatchNo (None, 35, 35, 32) 96 conv2d_37[0][0]

batch_normalization_39 (BatchNo (None, 35, 35, 32) 96 conv2d_39[0][0]

batch_normalization_42 (BatchNo (None, 35, 35, 64) 192 conv2d_42[0][0]

activation_37 (Activation) (None, 35, 35, 32) 0 batch_normalization_37[0][0]

activation_39 (Activation) (None, 35, 35, 32) 0 batch_normalization_39[0][0]

activation_42 (Activation) (None, 35, 35, 64) 0 batch_normalization_42[0][0]

block35_5_mixed (Concatenate) (None, 35, 35, 128) 0 activation_37[0][0]
activation_39[0][0]
activation_42[0][0]

block35_5_conv (Conv2D) (None, 35, 35, 320) 41280 block35_5_mixed[0][0]

block35_5 (Lambda) (None, 35, 35, 320) 0 block35_4_ac[0][0]
block35_5_conv[0][0]

block35_5_ac (Activation) (None, 35, 35, 320) 0 block35_5[0][0]

conv2d_46 (Conv2D) (None, 35, 35, 32) 10240 block35_5_ac[0][0]

batch_normalization_46 (BatchNo (None, 35, 35, 32) 96 conv2d_46[0][0]

activation_46 (Activation) (None, 35, 35, 32) 0 batch_normalization_46[0][0]

conv2d_44 (Conv2D) (None, 35, 35, 32) 10240 block35_5_ac[0][0]

conv2d_47 (Conv2D) (None, 35, 35, 48) 13824 activation_46[0][0]

batch_normalization_44 (BatchNo (None, 35, 35, 32) 96 conv2d_44[0][0]

batch_normalization_47 (BatchNo (None, 35, 35, 48) 144 conv2d_47[0][0]

activation_44 (Activation) (None, 35, 35, 32) 0 batch_normalization_44[0][0]

activation_47 (Activation) (None, 35, 35, 48) 0 batch_normalization_47[0][0]

conv2d_43 (Conv2D) (None, 35, 35, 32) 10240 block35_5_ac[0][0]

conv2d_45 (Conv2D) (None, 35, 35, 32) 9216 activation_44[0][0]

conv2d_48 (Conv2D) (None, 35, 35, 64) 27648 activation_47[0][0]

batch_normalization_43 (BatchNo (None, 35, 35, 32) 96 conv2d_43[0][0]

batch_normalization_45 (BatchNo (None, 35, 35, 32) 96 conv2d_45[0][0]

batch_normalization_48 (BatchNo (None, 35, 35, 64) 192 conv2d_48[0][0]

activation_43 (Activation) (None, 35, 35, 32) 0 batch_normalization_43[0][0]

activation_45 (Activation) (None, 35, 35, 32) 0 batch_normalization_45[0][0]

activation_48 (Activation) (None, 35, 35, 64) 0 batch_normalization_48[0][0]

block35_6_mixed (Concatenate) (None, 35, 35, 128) 0 activation_43[0][0]
activation_45[0][0]
activation_48[0][0]

block35_6_conv (Conv2D) (None, 35, 35, 320) 41280 block35_6_mixed[0][0]

block35_6 (Lambda) (None, 35, 35, 320) 0 block35_5_ac[0][0]
block35_6_conv[0][0]

block35_6_ac (Activation) (None, 35, 35, 320) 0 block35_6[0][0]

conv2d_52 (Conv2D) (None, 35, 35, 32) 10240 block35_6_ac[0][0]

batch_normalization_52 (BatchNo (None, 35, 35, 32) 96 conv2d_52[0][0]

activation_52 (Activation) (None, 35, 35, 32) 0 batch_normalization_52[0][0]

conv2d_50 (Conv2D) (None, 35, 35, 32) 10240 block35_6_ac[0][0]

conv2d_53 (Conv2D) (None, 35, 35, 48) 13824 activation_52[0][0]

batch_normalization_50 (BatchNo (None, 35, 35, 32) 96 conv2d_50[0][0]

batch_normalization_53 (BatchNo (None, 35, 35, 48) 144 conv2d_53[0][0]

activation_50 (Activation) (None, 35, 35, 32) 0 batch_normalization_50[0][0]

activation_53 (Activation) (None, 35, 35, 48) 0 batch_normalization_53[0][0]

conv2d_49 (Conv2D) (None, 35, 35, 32) 10240 block35_6_ac[0][0]

conv2d_51 (Conv2D) (None, 35, 35, 32) 9216 activation_50[0][0]

conv2d_54 (Conv2D) (None, 35, 35, 64) 27648 activation_53[0][0]

batch_normalization_49 (BatchNo (None, 35, 35, 32) 96 conv2d_49[0][0]

batch_normalization_51 (BatchNo (None, 35, 35, 32) 96 conv2d_51[0][0]

batch_normalization_54 (BatchNo (None, 35, 35, 64) 192 conv2d_54[0][0]

activation_49 (Activation) (None, 35, 35, 32) 0 batch_normalization_49[0][0]

activation_51 (Activation) (None, 35, 35, 32) 0 batch_normalization_51[0][0]

activation_54 (Activation) (None, 35, 35, 64) 0 batch_normalization_54[0][0]

block35_7_mixed (Concatenate) (None, 35, 35, 128) 0 activation_49[0][0]
activation_51[0][0]
activation_54[0][0]

block35_7_conv (Conv2D) (None, 35, 35, 320) 41280 block35_7_mixed[0][0]

block35_7 (Lambda) (None, 35, 35, 320) 0 block35_6_ac[0][0]
block35_7_conv[0][0]

block35_7_ac (Activation) (None, 35, 35, 320) 0 block35_7[0][0]

conv2d_58 (Conv2D) (None, 35, 35, 32) 10240 block35_7_ac[0][0]

batch_normalization_58 (BatchNo (None, 35, 35, 32) 96 conv2d_58[0][0]

activation_58 (Activation) (None, 35, 35, 32) 0 batch_normalization_58[0][0]

conv2d_56 (Conv2D) (None, 35, 35, 32) 10240 block35_7_ac[0][0]

conv2d_59 (Conv2D) (None, 35, 35, 48) 13824 activation_58[0][0]

batch_normalization_56 (BatchNo (None, 35, 35, 32) 96 conv2d_56[0][0]

batch_normalization_59 (BatchNo (None, 35, 35, 48) 144 conv2d_59[0][0]

activation_56 (Activation) (None, 35, 35, 32) 0 batch_normalization_56[0][0]

activation_59 (Activation) (None, 35, 35, 48) 0 batch_normalization_59[0][0]

conv2d_55 (Conv2D) (None, 35, 35, 32) 10240 block35_7_ac[0][0]

conv2d_57 (Conv2D) (None, 35, 35, 32) 9216 activation_56[0][0]

conv2d_60 (Conv2D) (None, 35, 35, 64) 27648 activation_59[0][0]

batch_normalization_55 (BatchNo (None, 35, 35, 32) 96 conv2d_55[0][0]

batch_normalization_57 (BatchNo (None, 35, 35, 32) 96 conv2d_57[0][0]

batch_normalization_60 (BatchNo (None, 35, 35, 64) 192 conv2d_60[0][0]

activation_55 (Activation) (None, 35, 35, 32) 0 batch_normalization_55[0][0]

activation_57 (Activation) (None, 35, 35, 32) 0 batch_normalization_57[0][0]

activation_60 (Activation) (None, 35, 35, 64) 0 batch_normalization_60[0][0]

block35_8_mixed (Concatenate) (None, 35, 35, 128) 0 activation_55[0][0]
activation_57[0][0]
activation_60[0][0]

block35_8_conv (Conv2D) (None, 35, 35, 320) 41280 block35_8_mixed[0][0]

block35_8 (Lambda) (None, 35, 35, 320) 0 block35_7_ac[0][0]
block35_8_conv[0][0]

block35_8_ac (Activation) (None, 35, 35, 320) 0 block35_8[0][0]

conv2d_64 (Conv2D) (None, 35, 35, 32) 10240 block35_8_ac[0][0]

batch_normalization_64 (BatchNo (None, 35, 35, 32) 96 conv2d_64[0][0]

activation_64 (Activation) (None, 35, 35, 32) 0 batch_normalization_64[0][0]

conv2d_62 (Conv2D) (None, 35, 35, 32) 10240 block35_8_ac[0][0]

conv2d_65 (Conv2D) (None, 35, 35, 48) 13824 activation_64[0][0]

batch_normalization_62 (BatchNo (None, 35, 35, 32) 96 conv2d_62[0][0]

batch_normalization_65 (BatchNo (None, 35, 35, 48) 144 conv2d_65[0][0]

activation_62 (Activation) (None, 35, 35, 32) 0 batch_normalization_62[0][0]

activation_65 (Activation) (None, 35, 35, 48) 0 batch_normalization_65[0][0]

conv2d_61 (Conv2D) (None, 35, 35, 32) 10240 block35_8_ac[0][0]

conv2d_63 (Conv2D) (None, 35, 35, 32) 9216 activation_62[0][0]

conv2d_66 (Conv2D) (None, 35, 35, 64) 27648 activation_65[0][0]

batch_normalization_61 (BatchNo (None, 35, 35, 32) 96 conv2d_61[0][0]

batch_normalization_63 (BatchNo (None, 35, 35, 32) 96 conv2d_63[0][0]

batch_normalization_66 (BatchNo (None, 35, 35, 64) 192 conv2d_66[0][0]

activation_61 (Activation) (None, 35, 35, 32) 0 batch_normalization_61[0][0]

activation_63 (Activation) (None, 35, 35, 32) 0 batch_normalization_63[0][0]

activation_66 (Activation) (None, 35, 35, 64) 0 batch_normalization_66[0][0]

block35_9_mixed (Concatenate) (None, 35, 35, 128) 0 activation_61[0][0]
activation_63[0][0]
activation_66[0][0]

block35_9_conv (Conv2D) (None, 35, 35, 320) 41280 block35_9_mixed[0][0]

block35_9 (Lambda) (None, 35, 35, 320) 0 block35_8_ac[0][0]
block35_9_conv[0][0]

block35_9_ac (Activation) (None, 35, 35, 320) 0 block35_9[0][0]

conv2d_70 (Conv2D) (None, 35, 35, 32) 10240 block35_9_ac[0][0]

batch_normalization_70 (BatchNo (None, 35, 35, 32) 96 conv2d_70[0][0]

activation_70 (Activation) (None, 35, 35, 32) 0 batch_normalization_70[0][0]

conv2d_68 (Conv2D) (None, 35, 35, 32) 10240 block35_9_ac[0][0]

conv2d_71 (Conv2D) (None, 35, 35, 48) 13824 activation_70[0][0]

batch_normalization_68 (BatchNo (None, 35, 35, 32) 96 conv2d_68[0][0]

batch_normalization_71 (BatchNo (None, 35, 35, 48) 144 conv2d_71[0][0]

activation_68 (Activation) (None, 35, 35, 32) 0 batch_normalization_68[0][0]

activation_71 (Activation) (None, 35, 35, 48) 0 batch_normalization_71[0][0]

conv2d_67 (Conv2D) (None, 35, 35, 32) 10240 block35_9_ac[0][0]

conv2d_69 (Conv2D) (None, 35, 35, 32) 9216 activation_68[0][0]

conv2d_72 (Conv2D) (None, 35, 35, 64) 27648 activation_71[0][0]

batch_normalization_67 (BatchNo (None, 35, 35, 32) 96 conv2d_67[0][0]

batch_normalization_69 (BatchNo (None, 35, 35, 32) 96 conv2d_69[0][0]

batch_normalization_72 (BatchNo (None, 35, 35, 64) 192 conv2d_72[0][0]

activation_67 (Activation) (None, 35, 35, 32) 0 batch_normalization_67[0][0]

activation_69 (Activation) (None, 35, 35, 32) 0 batch_normalization_69[0][0]

activation_72 (Activation) (None, 35, 35, 64) 0 batch_normalization_72[0][0]

block35_10_mixed (Concatenate) (None, 35, 35, 128) 0 activation_67[0][0]
activation_69[0][0]
activation_72[0][0]

block35_10_conv (Conv2D) (None, 35, 35, 320) 41280 block35_10_mixed[0][0]

block35_10 (Lambda) (None, 35, 35, 320) 0 block35_9_ac[0][0]
block35_10_conv[0][0]

block35_10_ac (Activation) (None, 35, 35, 320) 0 block35_10[0][0]

conv2d_74 (Conv2D) (None, 35, 35, 256) 81920 block35_10_ac[0][0]

batch_normalization_74 (BatchNo (None, 35, 35, 256) 768 conv2d_74[0][0]

activation_74 (Activation) (None, 35, 35, 256) 0 batch_normalization_74[0][0]

conv2d_75 (Conv2D) (None, 35, 35, 256) 589824 activation_74[0][0]

batch_normalization_75 (BatchNo (None, 35, 35, 256) 768 conv2d_75[0][0]

activation_75 (Activation) (None, 35, 35, 256) 0 batch_normalization_75[0][0]

conv2d_73 (Conv2D) (None, 17, 17, 384) 1105920 block35_10_ac[0][0]

conv2d_76 (Conv2D) (None, 17, 17, 384) 884736 activation_75[0][0]

batch_normalization_73 (BatchNo (None, 17, 17, 384) 1152 conv2d_73[0][0]

batch_normalization_76 (BatchNo (None, 17, 17, 384) 1152 conv2d_76[0][0]

activation_73 (Activation) (None, 17, 17, 384) 0 batch_normalization_73[0][0]

activation_76 (Activation) (None, 17, 17, 384) 0 batch_normalization_76[0][0]

max_pooling2d_3 (MaxPooling2D) (None, 17, 17, 320) 0 block35_10_ac[0][0]

mixed_6a (Concatenate) (None, 17, 17, 1088) 0 activation_73[0][0]
activation_76[0][0]
max_pooling2d_3[0][0]

conv2d_78 (Conv2D) (None, 17, 17, 128) 139264 mixed_6a[0][0]

batch_normalization_78 (BatchNo (None, 17, 17, 128) 384 conv2d_78[0][0]

activation_78 (Activation) (None, 17, 17, 128) 0 batch_normalization_78[0][0]

conv2d_79 (Conv2D) (None, 17, 17, 160) 143360 activation_78[0][0]

batch_normalization_79 (BatchNo (None, 17, 17, 160) 480 conv2d_79[0][0]

activation_79 (Activation) (None, 17, 17, 160) 0 batch_normalization_79[0][0]

conv2d_77 (Conv2D) (None, 17, 17, 192) 208896 mixed_6a[0][0]

conv2d_80 (Conv2D) (None, 17, 17, 192) 215040 activation_79[0][0]

batch_normalization_77 (BatchNo (None, 17, 17, 192) 576 conv2d_77[0][0]

batch_normalization_80 (BatchNo (None, 17, 17, 192) 576 conv2d_80[0][0]

activation_77 (Activation) (None, 17, 17, 192) 0 batch_normalization_77[0][0]

activation_80 (Activation) (None, 17, 17, 192) 0 batch_normalization_80[0][0]

block17_1_mixed (Concatenate) (None, 17, 17, 384) 0 activation_77[0][0]
activation_80[0][0]

block17_1_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_1_mixed[0][0]

block17_1 (Lambda) (None, 17, 17, 1088) 0 mixed_6a[0][0]
block17_1_conv[0][0]

block17_1_ac (Activation) (None, 17, 17, 1088) 0 block17_1[0][0]

conv2d_82 (Conv2D) (None, 17, 17, 128) 139264 block17_1_ac[0][0]

batch_normalization_82 (BatchNo (None, 17, 17, 128) 384 conv2d_82[0][0]

activation_82 (Activation) (None, 17, 17, 128) 0 batch_normalization_82[0][0]

conv2d_83 (Conv2D) (None, 17, 17, 160) 143360 activation_82[0][0]

batch_normalization_83 (BatchNo (None, 17, 17, 160) 480 conv2d_83[0][0]

activation_83 (Activation) (None, 17, 17, 160) 0 batch_normalization_83[0][0]

conv2d_81 (Conv2D) (None, 17, 17, 192) 208896 block17_1_ac[0][0]

conv2d_84 (Conv2D)	(None, 17, 17, 192)	215040	activation_83[0][0]
batch_normalization_81 (BatchNo	(None, 17, 17, 192)	576	conv2d_81[0][0]
batch_normalization_84 (BatchNo	(None, 17, 17, 192)	576	conv2d_84[0][0]
activation_81 (Activation)	(None, 17, 17, 192)	0	batch_normalization_81[0][0]
activation_84 (Activation)	(None, 17, 17, 192)	0	batch_normalization_84[0][0]
block17_2_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_81[0][0] activation_84[0][0]
block17_2_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_2_mixed[0][0]
block17_2 (Lambda)	(None, 17, 17, 1088)	0	block17_1_ac[0][0] block17_2_conv[0][0]
block17_2_ac (Activation)	(None, 17, 17, 1088)	0	block17_2[0][0]
conv2d_86 (Conv2D)	(None, 17, 17, 128)	139264	block17_2_ac[0][0]
batch_normalization_86 (BatchNo	(None, 17, 17, 128)	384	conv2d_86[0][0]
activation_86 (Activation)	(None, 17, 17, 128)	0	batch_normalization_86[0][0]
conv2d_87 (Conv2D)	(None, 17, 17, 160)	143360	activation_86[0][0]
batch_normalization_87 (BatchNo	(None, 17, 17, 160)	480	conv2d_87[0][0]
activation_87 (Activation)	(None, 17, 17, 160)	0	batch_normalization_87[0][0]
conv2d_85 (Conv2D)	(None, 17, 17, 192)	208896	block17_2_ac[0][0]
conv2d_88 (Conv2D)	(None, 17, 17, 192)	215040	activation_87[0][0]
batch_normalization_85 (BatchNo	(None, 17, 17, 192)	576	conv2d_85[0][0]
batch_normalization_88 (BatchNo	(None, 17, 17, 192)	576	conv2d_88[0][0]
activation_85 (Activation)	(None, 17, 17, 192)	0	batch_normalization_85[0][0]
activation_88 (Activation)	(None, 17, 17, 192)	0	batch_normalization_88[0][0]
block17_3_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_85[0][0] activation_88[0][0]
block17_3_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_3_mixed[0][0]
block17_3 (Lambda)	(None, 17, 17, 1088)	0	block17_2_ac[0][0] block17_3_conv[0][0]
block17_3_ac (Activation)	(None, 17, 17, 1088)	0	block17_3[0][0]
conv2d_90 (Conv2D)	(None, 17, 17, 128)	139264	block17_3_ac[0][0]
batch_normalization_90 (BatchNo	(None, 17, 17, 128)	384	conv2d_90[0][0]
activation_90 (Activation)	(None, 17, 17, 128)	0	batch_normalization_90[0][0]
conv2d_91 (Conv2D)	(None, 17, 17, 160)	143360	activation_90[0][0]
batch_normalization_91 (BatchNo	(None, 17, 17, 160)	480	conv2d_91[0][0]
activation_91 (Activation)	(None, 17, 17, 160)	0	batch_normalization_91[0][0]
conv2d_89 (Conv2D)	(None, 17, 17, 192)	208896	block17_3_ac[0][0]
conv2d_92 (Conv2D)	(None, 17, 17, 192)	215040	activation_91[0][0]
batch_normalization_89 (BatchNo	(None, 17, 17, 192)	576	conv2d_89[0][0]
batch_normalization_92 (BatchNo	(None, 17, 17, 192)	576	conv2d_92[0][0]
activation_89 (Activation)	(None, 17, 17, 192)	0	batch_normalization_89[0][0]
activation_92 (Activation)	(None, 17, 17, 192)	0	batch_normalization_92[0][0]
block17_4_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_89[0][0] activation_92[0][0]
block17_4_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_4_mixed[0][0]

block17_4 (Lambda) (None, 17, 17, 1088) 0 block17_3_ac[0][0]
block17_4_conv[0][0]

block17_4_ac (Activation) (None, 17, 17, 1088) 0 block17_4[0][0]

conv2d_94 (Conv2D) (None, 17, 17, 128) 139264 block17_4_ac[0][0]

batch_normalization_94 (BatchNo (None, 17, 17, 128) 384 conv2d_94[0][0]

activation_94 (Activation) (None, 17, 17, 128) 0 batch_normalization_94[0][0]

conv2d_95 (Conv2D) (None, 17, 17, 160) 143360 activation_94[0][0]

batch_normalization_95 (BatchNo (None, 17, 17, 160) 480 conv2d_95[0][0]

activation_95 (Activation) (None, 17, 17, 160) 0 batch_normalization_95[0][0]

conv2d_93 (Conv2D) (None, 17, 17, 192) 208896 block17_4_ac[0][0]

conv2d_96 (Conv2D) (None, 17, 17, 192) 215040 activation_95[0][0]

batch_normalization_93 (BatchNo (None, 17, 17, 192) 576 conv2d_93[0][0]

batch_normalization_96 (BatchNo (None, 17, 17, 192) 576 conv2d_96[0][0]

activation_93 (Activation) (None, 17, 17, 192) 0 batch_normalization_93[0][0]

activation_96 (Activation) (None, 17, 17, 192) 0 batch_normalization_96[0][0]

block17_5_mixed (Concatenate) (None, 17, 17, 384) 0 activation_93[0][0]
activation_96[0][0]

block17_5_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_5_mixed[0][0]

block17_5 (Lambda) (None, 17, 17, 1088) 0 block17_4_ac[0][0]
block17_5_conv[0][0]

block17_5_ac (Activation) (None, 17, 17, 1088) 0 block17_5[0][0]

conv2d_98 (Conv2D) (None, 17, 17, 128) 139264 block17_5_ac[0][0]

batch_normalization_98 (BatchNo (None, 17, 17, 128) 384 conv2d_98[0][0]

activation_98 (Activation) (None, 17, 17, 128) 0 batch_normalization_98[0][0]

conv2d_99 (Conv2D) (None, 17, 17, 160) 143360 activation_98[0][0]

batch_normalization_99 (BatchNo (None, 17, 17, 160) 480 conv2d_99[0][0]

activation_99 (Activation) (None, 17, 17, 160) 0 batch_normalization_99[0][0]

conv2d_97 (Conv2D) (None, 17, 17, 192) 208896 block17_5_ac[0][0]

conv2d_100 (Conv2D) (None, 17, 17, 192) 215040 activation_99[0][0]

batch_normalization_97 (BatchNo (None, 17, 17, 192) 576 conv2d_97[0][0]

batch_normalization_100 (BatchN (None, 17, 17, 192) 576 conv2d_100[0][0]

activation_97 (Activation) (None, 17, 17, 192) 0 batch_normalization_97[0][0]

activation_100 (Activation) (None, 17, 17, 192) 0 batch_normalization_100[0][0]

block17_6_mixed (Concatenate) (None, 17, 17, 384) 0 activation_97[0][0]
activation_100[0][0]

block17_6_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_6_mixed[0][0]

block17_6 (Lambda) (None, 17, 17, 1088) 0 block17_5_ac[0][0]
block17_6_conv[0][0]

block17_6_ac (Activation) (None, 17, 17, 1088) 0 block17_6[0][0]

conv2d_102 (Conv2D) (None, 17, 17, 128) 139264 block17_6_ac[0][0]

batch_normalization_102 (BatchN (None, 17, 17, 128) 384 conv2d_102[0][0]

activation_102 (Activation) (None, 17, 17, 128) 0 batch_normalization_102[0][0]

conv2d_103 (Conv2D) (None, 17, 17, 160) 143360 activation_102[0][0]

batch_normalization_103 (BatchN (None, 17, 17, 160) 480 conv2d_103[0][0]

activation_103 (Activation) (None, 17, 17, 160) 0 batch_normalization_103[0][0]

conv2d_101 (Conv2D) (None, 17, 17, 192) 208896 block17_6_ac[0][0]

conv2d_104 (Conv2D) (None, 17, 17, 192) 215040 activation_103[0][0]

batch_normalization_101 (BatchN (None, 17, 17, 192) 576 conv2d_101[0][0]

batch_normalization_104 (BatchN (None, 17, 17, 192) 576 conv2d_104[0][0]

activation_101 (Activation) (None, 17, 17, 192) 0 batch_normalization_101[0][0]

activation_104 (Activation) (None, 17, 17, 192) 0 batch_normalization_104[0][0]

block17_7_mixed (Concatenate) (None, 17, 17, 384) 0 activation_101[0][0]
activation_104[0][0]

block17_7_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_7_mixed[0][0]

block17_7 (Lambda) (None, 17, 17, 1088) 0 block17_6_ac[0][0]
block17_7_conv[0][0]

block17_7_ac (Activation) (None, 17, 17, 1088) 0 block17_7[0][0]

conv2d_106 (Conv2D) (None, 17, 17, 128) 139264 block17_7_ac[0][0]

batch_normalization_106 (BatchN (None, 17, 17, 128) 384 conv2d_106[0][0]

activation_106 (Activation) (None, 17, 17, 128) 0 batch_normalization_106[0][0]

conv2d_107 (Conv2D) (None, 17, 17, 160) 143360 activation_106[0][0]

batch_normalization_107 (BatchN (None, 17, 17, 160) 480 conv2d_107[0][0]

activation_107 (Activation) (None, 17, 17, 160) 0 batch_normalization_107[0][0]

conv2d_105 (Conv2D) (None, 17, 17, 192) 208896 block17_7_ac[0][0]

conv2d_108 (Conv2D) (None, 17, 17, 192) 215040 activation_107[0][0]

batch_normalization_105 (BatchN (None, 17, 17, 192) 576 conv2d_105[0][0]

batch_normalization_108 (BatchN (None, 17, 17, 192) 576 conv2d_108[0][0]

activation_105 (Activation) (None, 17, 17, 192) 0 batch_normalization_105[0][0]

activation_108 (Activation) (None, 17, 17, 192) 0 batch_normalization_108[0][0]

block17_8_mixed (Concatenate) (None, 17, 17, 384) 0 activation_105[0][0]
activation_108[0][0]

block17_8_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_8_mixed[0][0]

block17_8 (Lambda) (None, 17, 17, 1088) 0 block17_7_ac[0][0]
block17_8_conv[0][0]

block17_8_ac (Activation) (None, 17, 17, 1088) 0 block17_8[0][0]

conv2d_110 (Conv2D) (None, 17, 17, 128) 139264 block17_8_ac[0][0]

batch_normalization_110 (BatchN (None, 17, 17, 128) 384 conv2d_110[0][0]

activation_110 (Activation) (None, 17, 17, 128) 0 batch_normalization_110[0][0]

conv2d_111 (Conv2D) (None, 17, 17, 160) 143360 activation_110[0][0]

batch_normalization_111 (BatchN (None, 17, 17, 160) 480 conv2d_111[0][0]

activation_111 (Activation) (None, 17, 17, 160) 0 batch_normalization_111[0][0]

conv2d_109 (Conv2D) (None, 17, 17, 192) 208896 block17_8_ac[0][0]

conv2d_112 (Conv2D) (None, 17, 17, 192) 215040 activation_111[0][0]

batch_normalization_109 (BatchN (None, 17, 17, 192) 576 conv2d_109[0][0]

batch_normalization_112 (BatchN (None, 17, 17, 192) 576 conv2d_112[0][0]

activation_109 (Activation) (None, 17, 17, 192) 0 batch_normalization_109[0][0]

activation_112 (Activation) (None, 17, 17, 192) 0 batch_normalization_112[0][0]

block17_9_mixed (Concatenate) (None, 17, 17, 384) 0 activation_109[0][0]
activation_112[0][0]

block17_9_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_9_mixed[0][0]

block17_9 (Lambda) (None, 17, 17, 1088) 0 block17_8_ac[0][0]
block17_9_conv[0][0]

block17_9_ac (Activation) (None, 17, 17, 1088) 0 block17_9[0][0]

conv2d_114 (Conv2D) (None, 17, 17, 128) 139264 block17_9_ac[0][0]

batch_normalization_114 (BatchN (None, 17, 17, 128) 384 conv2d_114[0][0]

activation_114 (Activation) (None, 17, 17, 128) 0 batch_normalization_114[0][0]

conv2d_115 (Conv2D) (None, 17, 17, 160) 143360 activation_114[0][0]

batch_normalization_115 (BatchN (None, 17, 17, 160) 480 conv2d_115[0][0]

activation_115 (Activation) (None, 17, 17, 160) 0 batch_normalization_115[0][0]

conv2d_113 (Conv2D) (None, 17, 17, 192) 208896 block17_9_ac[0][0]

conv2d_116 (Conv2D) (None, 17, 17, 192) 215040 activation_115[0][0]

batch_normalization_113 (BatchN (None, 17, 17, 192) 576 conv2d_113[0][0]

batch_normalization_116 (BatchN (None, 17, 17, 192) 576 conv2d_116[0][0]

activation_113 (Activation) (None, 17, 17, 192) 0 batch_normalization_113[0][0]

activation_116 (Activation) (None, 17, 17, 192) 0 batch_normalization_116[0][0]

block17_10_mixed (Concatenate) (None, 17, 17, 384) 0 activation_113[0][0]
activation_116[0][0]

block17_10_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_10_mixed[0][0]

block17_10 (Lambda) (None, 17, 17, 1088) 0 block17_9_ac[0][0]
block17_10_conv[0][0]

block17_10_ac (Activation) (None, 17, 17, 1088) 0 block17_10[0][0]

conv2d_118 (Conv2D) (None, 17, 17, 128) 139264 block17_10_ac[0][0]

batch_normalization_118 (BatchN (None, 17, 17, 128) 384 conv2d_118[0][0]

activation_118 (Activation) (None, 17, 17, 128) 0 batch_normalization_118[0][0]

conv2d_119 (Conv2D) (None, 17, 17, 160) 143360 activation_118[0][0]

batch_normalization_119 (BatchN (None, 17, 17, 160) 480 conv2d_119[0][0]

activation_119 (Activation) (None, 17, 17, 160) 0 batch_normalization_119[0][0]

conv2d_117 (Conv2D) (None, 17, 17, 192) 208896 block17_10_ac[0][0]

conv2d_120 (Conv2D) (None, 17, 17, 192) 215040 activation_119[0][0]

batch_normalization_117 (BatchN (None, 17, 17, 192) 576 conv2d_117[0][0]

batch_normalization_120 (BatchN (None, 17, 17, 192) 576 conv2d_120[0][0]

activation_117 (Activation) (None, 17, 17, 192) 0 batch_normalization_117[0][0]

activation_120 (Activation) (None, 17, 17, 192) 0 batch_normalization_120[0][0]

block17_11_mixed (Concatenate) (None, 17, 17, 384) 0 activation_117[0][0]
activation_120[0][0]

block17_11_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_11_mixed[0][0]

block17_11 (Lambda) (None, 17, 17, 1088) 0 block17_10_ac[0][0]
block17_11_conv[0][0]

block17_11_ac (Activation) (None, 17, 17, 1088) 0 block17_11[0][0]

conv2d_122 (Conv2D) (None, 17, 17, 128) 139264 block17_11_ac[0][0]

batch_normalization_122 (BatchN (None, 17, 17, 128) 384 conv2d_122[0][0]

activation_122 (Activation) (None, 17, 17, 128) 0 batch_normalization_122[0][0]

conv2d_123 (Conv2D) (None, 17, 17, 160) 143360 activation_122[0][0]

batch_normalization_123 (BatchN (None, 17, 17, 160) 480 conv2d_123[0][0]

activation_123 (Activation) (None, 17, 17, 160) 0 batch_normalization_123[0][0]

conv2d_121 (Conv2D) (None, 17, 17, 192) 208896 block17_11_ac[0][0]

conv2d_124 (Conv2D) (None, 17, 17, 192) 215040 activation_123[0][0]

batch_normalization_121 (BatchN (None, 17, 17, 192) 576 conv2d_121[0][0]

batch_normalization_124 (BatchN (None, 17, 17, 192) 576 conv2d_124[0][0]

activation_121 (Activation) (None, 17, 17, 192) 0 batch_normalization_121[0][0]

activation_124 (Activation) (None, 17, 17, 192) 0 batch_normalization_124[0][0]

block17_12_mixed (Concatenate) (None, 17, 17, 384) 0 activation_121[0][0]
activation_124[0][0]

block17_12_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_12_mixed[0][0]

block17_12 (Lambda) (None, 17, 17, 1088) 0 block17_11_ac[0][0]
block17_12_conv[0][0]

block17_12_ac (Activation) (None, 17, 17, 1088) 0 block17_12[0][0]

conv2d_126 (Conv2D) (None, 17, 17, 128) 139264 block17_12_ac[0][0]

batch_normalization_126 (BatchN (None, 17, 17, 128) 384 conv2d_126[0][0]

activation_126 (Activation) (None, 17, 17, 128) 0 batch_normalization_126[0][0]

conv2d_127 (Conv2D) (None, 17, 17, 160) 143360 activation_126[0][0]

batch_normalization_127 (BatchN (None, 17, 17, 160) 480 conv2d_127[0][0]

activation_127 (Activation) (None, 17, 17, 160) 0 batch_normalization_127[0][0]

conv2d_125 (Conv2D) (None, 17, 17, 192) 208896 block17_12_ac[0][0]

conv2d_128 (Conv2D) (None, 17, 17, 192) 215040 activation_127[0][0]

batch_normalization_125 (BatchN (None, 17, 17, 192) 576 conv2d_125[0][0]

batch_normalization_128 (BatchN (None, 17, 17, 192) 576 conv2d_128[0][0]

activation_125 (Activation) (None, 17, 17, 192) 0 batch_normalization_125[0][0]

activation_128 (Activation) (None, 17, 17, 192) 0 batch_normalization_128[0][0]

block17_13_mixed (Concatenate) (None, 17, 17, 384) 0 activation_125[0][0]
activation_128[0][0]

block17_13_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_13_mixed[0][0]

block17_13 (Lambda) (None, 17, 17, 1088) 0 block17_12_ac[0][0]
block17_13_conv[0][0]

block17_13_ac (Activation) (None, 17, 17, 1088) 0 block17_13[0][0]

conv2d_130 (Conv2D) (None, 17, 17, 128) 139264 block17_13_ac[0][0]

batch_normalization_130 (BatchN (None, 17, 17, 128) 384 conv2d_130[0][0]

activation_130 (Activation) (None, 17, 17, 128) 0 batch_normalization_130[0][0]

conv2d_131 (Conv2D) (None, 17, 17, 160) 143360 activation_130[0][0]

batch_normalization_131 (BatchN (None, 17, 17, 160) 480 conv2d_131[0][0]

activation_131 (Activation) (None, 17, 17, 160) 0 batch_normalization_131[0][0]

conv2d_129 (Conv2D) (None, 17, 17, 192) 208896 block17_13_ac[0][0]

conv2d_132 (Conv2D) (None, 17, 17, 192) 215040 activation_131[0][0]

batch_normalization_129 (BatchN (None, 17, 17, 192) 576 conv2d_129[0][0]

batch_normalization_132 (BatchN (None, 17, 17, 192) 576 conv2d_132[0][0]

activation_129 (Activation) (None, 17, 17, 192) 0 batch_normalization_129[0][0]

activation_132 (Activation) (None, 17, 17, 192) 0 batch_normalization_132[0][0]

block17_14_mixed (Concatenate) (None, 17, 17, 384) 0 activation_129[0][0]

			activation_132[0][0]
block17_14_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_14_mixed[0][0]
block17_14 (Lambda)	(None, 17, 17, 1088)	0	block17_13_ac[0][0] block17_14_conv[0][0]
block17_14_ac (Activation)	(None, 17, 17, 1088)	0	block17_14[0][0]
conv2d_134 (Conv2D)	(None, 17, 17, 128)	139264	block17_14_ac[0][0]
batch_normalization_134 (BatchN	(None, 17, 17, 128)	384	conv2d_134[0][0]
activation_134 (Activation)	(None, 17, 17, 128)	0	batch_normalization_134[0][0]
conv2d_135 (Conv2D)	(None, 17, 17, 160)	143360	activation_134[0][0]
batch_normalization_135 (BatchN	(None, 17, 17, 160)	480	conv2d_135[0][0]
activation_135 (Activation)	(None, 17, 17, 160)	0	batch_normalization_135[0][0]
conv2d_133 (Conv2D)	(None, 17, 17, 192)	208896	block17_14_ac[0][0]
conv2d_136 (Conv2D)	(None, 17, 17, 192)	215040	activation_135[0][0]
batch_normalization_133 (BatchN	(None, 17, 17, 192)	576	conv2d_133[0][0]
batch_normalization_136 (BatchN	(None, 17, 17, 192)	576	conv2d_136[0][0]
activation_133 (Activation)	(None, 17, 17, 192)	0	batch_normalization_133[0][0]
activation_136 (Activation)	(None, 17, 17, 192)	0	batch_normalization_136[0][0]
block17_15_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_133[0][0] activation_136[0][0]
block17_15_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_15_mixed[0][0]
block17_15 (Lambda)	(None, 17, 17, 1088)	0	block17_14_ac[0][0] block17_15_conv[0][0]
block17_15_ac (Activation)	(None, 17, 17, 1088)	0	block17_15[0][0]
conv2d_138 (Conv2D)	(None, 17, 17, 128)	139264	block17_15_ac[0][0]
batch_normalization_138 (BatchN	(None, 17, 17, 128)	384	conv2d_138[0][0]
activation_138 (Activation)	(None, 17, 17, 128)	0	batch_normalization_138[0][0]
conv2d_139 (Conv2D)	(None, 17, 17, 160)	143360	activation_138[0][0]
batch_normalization_139 (BatchN	(None, 17, 17, 160)	480	conv2d_139[0][0]
activation_139 (Activation)	(None, 17, 17, 160)	0	batch_normalization_139[0][0]
conv2d_137 (Conv2D)	(None, 17, 17, 192)	208896	block17_15_ac[0][0]
conv2d_140 (Conv2D)	(None, 17, 17, 192)	215040	activation_139[0][0]
batch_normalization_137 (BatchN	(None, 17, 17, 192)	576	conv2d_137[0][0]
batch_normalization_140 (BatchN	(None, 17, 17, 192)	576	conv2d_140[0][0]
activation_137 (Activation)	(None, 17, 17, 192)	0	batch_normalization_137[0][0]
activation_140 (Activation)	(None, 17, 17, 192)	0	batch_normalization_140[0][0]
block17_16_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_137[0][0] activation_140[0][0]
block17_16_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_16_mixed[0][0]
block17_16 (Lambda)	(None, 17, 17, 1088)	0	block17_15_ac[0][0] block17_16_conv[0][0]
block17_16_ac (Activation)	(None, 17, 17, 1088)	0	block17_16[0][0]
conv2d_142 (Conv2D)	(None, 17, 17, 128)	139264	block17_16_ac[0][0]
batch_normalization_142 (BatchN	(None, 17, 17, 128)	384	conv2d_142[0][0]
activation_142 (Activation)	(None, 17, 17, 128)	0	batch_normalization_142[0][0]
conv2d_143 (Conv2D)	(None, 17, 17, 160)	143360	activation_142[0][0]

batch_normalization_143 (BatchN (None, 17, 17, 160) 480 conv2d_143[0][0]

activation_143 (Activation) (None, 17, 17, 160) 0 batch_normalization_143[0][0]

conv2d_141 (Conv2D) (None, 17, 17, 192) 208896 block17_16_ac[0][0]

conv2d_144 (Conv2D) (None, 17, 17, 192) 215040 activation_143[0][0]

batch_normalization_141 (BatchN (None, 17, 17, 192) 576 conv2d_141[0][0]

batch_normalization_144 (BatchN (None, 17, 17, 192) 576 conv2d_144[0][0]

activation_141 (Activation) (None, 17, 17, 192) 0 batch_normalization_141[0][0]

activation_144 (Activation) (None, 17, 17, 192) 0 batch_normalization_144[0][0]

block17_17_mixed (Concatenate) (None, 17, 17, 384) 0 activation_141[0][0]
activation_144[0][0]

block17_17_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_17_mixed[0][0]

block17_17 (Lambda) (None, 17, 17, 1088) 0 block17_16_ac[0][0]
block17_17_conv[0][0]

block17_17_ac (Activation) (None, 17, 17, 1088) 0 block17_17[0][0]

conv2d_146 (Conv2D) (None, 17, 17, 128) 139264 block17_17_ac[0][0]

batch_normalization_146 (BatchN (None, 17, 17, 128) 384 conv2d_146[0][0]

activation_146 (Activation) (None, 17, 17, 128) 0 batch_normalization_146[0][0]

conv2d_147 (Conv2D) (None, 17, 17, 160) 143360 activation_146[0][0]

batch_normalization_147 (BatchN (None, 17, 17, 160) 480 conv2d_147[0][0]

activation_147 (Activation) (None, 17, 17, 160) 0 batch_normalization_147[0][0]

conv2d_145 (Conv2D) (None, 17, 17, 192) 208896 block17_17_ac[0][0]

conv2d_148 (Conv2D) (None, 17, 17, 192) 215040 activation_147[0][0]

batch_normalization_145 (BatchN (None, 17, 17, 192) 576 conv2d_145[0][0]

batch_normalization_148 (BatchN (None, 17, 17, 192) 576 conv2d_148[0][0]

activation_145 (Activation) (None, 17, 17, 192) 0 batch_normalization_145[0][0]

activation_148 (Activation) (None, 17, 17, 192) 0 batch_normalization_148[0][0]

block17_18_mixed (Concatenate) (None, 17, 17, 384) 0 activation_145[0][0]
activation_148[0][0]

block17_18_conv (Conv2D) (None, 17, 17, 1088) 418880 block17_18_mixed[0][0]

block17_18 (Lambda) (None, 17, 17, 1088) 0 block17_17_ac[0][0]
block17_18_conv[0][0]

block17_18_ac (Activation) (None, 17, 17, 1088) 0 block17_18[0][0]

conv2d_150 (Conv2D) (None, 17, 17, 128) 139264 block17_18_ac[0][0]

batch_normalization_150 (BatchN (None, 17, 17, 128) 384 conv2d_150[0][0]

activation_150 (Activation) (None, 17, 17, 128) 0 batch_normalization_150[0][0]

conv2d_151 (Conv2D) (None, 17, 17, 160) 143360 activation_150[0][0]

batch_normalization_151 (BatchN (None, 17, 17, 160) 480 conv2d_151[0][0]

activation_151 (Activation) (None, 17, 17, 160) 0 batch_normalization_151[0][0]

conv2d_149 (Conv2D) (None, 17, 17, 192) 208896 block17_18_ac[0][0]

conv2d_152 (Conv2D) (None, 17, 17, 192) 215040 activation_151[0][0]

batch_normalization_149 (BatchN (None, 17, 17, 192) 576 conv2d_149[0][0]

batch_normalization_152 (BatchN (None, 17, 17, 192) 576 conv2d_152[0][0]

activation_149 (Activation) (None, 17, 17, 192) 0 batch_normalization_149[0][0]

activation_152 (Activation) (None, 17, 17, 192) 0 batch_normalization_152[0][0]

block17_19_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_149[0][0] activation_152[0][0]
block17_19_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_19_mixed[0][0]
block17_19 (Lambda)	(None, 17, 17, 1088)	0	block17_18_ac[0][0] block17_19_conv[0][0]
block17_19_ac (Activation)	(None, 17, 17, 1088)	0	block17_19[0][0]
conv2d_154 (Conv2D)	(None, 17, 17, 128)	139264	block17_19_ac[0][0]
batch_normalization_154 (BatchN	(None, 17, 17, 128)	384	conv2d_154[0][0]
activation_154 (Activation)	(None, 17, 17, 128)	0	batch_normalization_154[0][0]
conv2d_155 (Conv2D)	(None, 17, 17, 160)	143360	activation_154[0][0]
batch_normalization_155 (BatchN	(None, 17, 17, 160)	480	conv2d_155[0][0]
activation_155 (Activation)	(None, 17, 17, 160)	0	batch_normalization_155[0][0]
conv2d_153 (Conv2D)	(None, 17, 17, 192)	208896	block17_19_ac[0][0]
conv2d_156 (Conv2D)	(None, 17, 17, 192)	215040	activation_155[0][0]
batch_normalization_153 (BatchN	(None, 17, 17, 192)	576	conv2d_153[0][0]
batch_normalization_156 (BatchN	(None, 17, 17, 192)	576	conv2d_156[0][0]
activation_153 (Activation)	(None, 17, 17, 192)	0	batch_normalization_153[0][0]
activation_156 (Activation)	(None, 17, 17, 192)	0	batch_normalization_156[0][0]
block17_20_mixed (Concatenate)	(None, 17, 17, 384)	0	activation_153[0][0] activation_156[0][0]
block17_20_conv (Conv2D)	(None, 17, 17, 1088)	418880	block17_20_mixed[0][0]
block17_20 (Lambda)	(None, 17, 17, 1088)	0	block17_19_ac[0][0] block17_20_conv[0][0]
block17_20_ac (Activation)	(None, 17, 17, 1088)	0	block17_20[0][0]
conv2d_161 (Conv2D)	(None, 17, 17, 256)	278528	block17_20_ac[0][0]
batch_normalization_161 (BatchN	(None, 17, 17, 256)	768	conv2d_161[0][0]
activation_161 (Activation)	(None, 17, 17, 256)	0	batch_normalization_161[0][0]
conv2d_157 (Conv2D)	(None, 17, 17, 256)	278528	block17_20_ac[0][0]
conv2d_159 (Conv2D)	(None, 17, 17, 256)	278528	block17_20_ac[0][0]
conv2d_162 (Conv2D)	(None, 17, 17, 288)	663552	activation_161[0][0]
batch_normalization_157 (BatchN	(None, 17, 17, 256)	768	conv2d_157[0][0]
batch_normalization_159 (BatchN	(None, 17, 17, 256)	768	conv2d_159[0][0]
batch_normalization_162 (BatchN	(None, 17, 17, 288)	864	conv2d_162[0][0]
activation_157 (Activation)	(None, 17, 17, 256)	0	batch_normalization_157[0][0]
activation_159 (Activation)	(None, 17, 17, 256)	0	batch_normalization_159[0][0]
activation_162 (Activation)	(None, 17, 17, 288)	0	batch_normalization_162[0][0]
conv2d_158 (Conv2D)	(None, 8, 8, 384)	884736	activation_157[0][0]
conv2d_160 (Conv2D)	(None, 8, 8, 288)	663552	activation_159[0][0]
conv2d_163 (Conv2D)	(None, 8, 8, 320)	829440	activation_162[0][0]
batch_normalization_158 (BatchN	(None, 8, 8, 384)	1152	conv2d_158[0][0]
batch_normalization_160 (BatchN	(None, 8, 8, 288)	864	conv2d_160[0][0]
batch_normalization_163 (BatchN	(None, 8, 8, 320)	960	conv2d_163[0][0]
activation_158 (Activation)	(None, 8, 8, 384)	0	batch_normalization_158[0][0]
activation_160 (Activation)	(None, 8, 8, 288)	0	batch_normalization_160[0][0]

activation_163 (Activation)	(None, 8, 8, 320)	0	batch_normalization_163[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 8, 8, 1088)	0	block17_20_ac[0][0]
mixed_7a (Concatenate)	(None, 8, 8, 2080)	0	activation_158[0][0] activation_160[0][0] activation_163[0][0] max_pooling2d_4[0][0]
conv2d_165 (Conv2D)	(None, 8, 8, 192)	399360	mixed_7a[0][0]
batch_normalization_165 (BatchN	(None, 8, 8, 192)	576	conv2d_165[0][0]
activation_165 (Activation)	(None, 8, 8, 192)	0	batch_normalization_165[0][0]
conv2d_166 (Conv2D)	(None, 8, 8, 224)	129024	activation_165[0][0]
batch_normalization_166 (BatchN	(None, 8, 8, 224)	672	conv2d_166[0][0]
activation_166 (Activation)	(None, 8, 8, 224)	0	batch_normalization_166[0][0]
conv2d_164 (Conv2D)	(None, 8, 8, 192)	399360	mixed_7a[0][0]
conv2d_167 (Conv2D)	(None, 8, 8, 256)	172032	activation_166[0][0]
batch_normalization_164 (BatchN	(None, 8, 8, 192)	576	conv2d_164[0][0]
batch_normalization_167 (BatchN	(None, 8, 8, 256)	768	conv2d_167[0][0]
activation_164 (Activation)	(None, 8, 8, 192)	0	batch_normalization_164[0][0]
activation_167 (Activation)	(None, 8, 8, 256)	0	batch_normalization_167[0][0]
block8_1_mixed (Concatenate)	(None, 8, 8, 448)	0	activation_164[0][0] activation_167[0][0]
block8_1_conv (Conv2D)	(None, 8, 8, 2080)	933920	block8_1_mixed[0][0]
block8_1 (Lambda)	(None, 8, 8, 2080)	0	mixed_7a[0][0] block8_1_conv[0][0]
block8_1_ac (Activation)	(None, 8, 8, 2080)	0	block8_1[0][0]
conv2d_169 (Conv2D)	(None, 8, 8, 192)	399360	block8_1_ac[0][0]
batch_normalization_169 (BatchN	(None, 8, 8, 192)	576	conv2d_169[0][0]
activation_169 (Activation)	(None, 8, 8, 192)	0	batch_normalization_169[0][0]
conv2d_170 (Conv2D)	(None, 8, 8, 224)	129024	activation_169[0][0]
batch_normalization_170 (BatchN	(None, 8, 8, 224)	672	conv2d_170[0][0]
activation_170 (Activation)	(None, 8, 8, 224)	0	batch_normalization_170[0][0]
conv2d_168 (Conv2D)	(None, 8, 8, 192)	399360	block8_1_ac[0][0]
conv2d_171 (Conv2D)	(None, 8, 8, 256)	172032	activation_170[0][0]
batch_normalization_168 (BatchN	(None, 8, 8, 192)	576	conv2d_168[0][0]
batch_normalization_171 (BatchN	(None, 8, 8, 256)	768	conv2d_171[0][0]
activation_168 (Activation)	(None, 8, 8, 192)	0	batch_normalization_168[0][0]
activation_171 (Activation)	(None, 8, 8, 256)	0	batch_normalization_171[0][0]
block8_2_mixed (Concatenate)	(None, 8, 8, 448)	0	activation_168[0][0] activation_171[0][0]
block8_2_conv (Conv2D)	(None, 8, 8, 2080)	933920	block8_2_mixed[0][0]
block8_2 (Lambda)	(None, 8, 8, 2080)	0	block8_1_ac[0][0] block8_2_conv[0][0]
block8_2_ac (Activation)	(None, 8, 8, 2080)	0	block8_2[0][0]
conv2d_173 (Conv2D)	(None, 8, 8, 192)	399360	block8_2_ac[0][0]
batch_normalization_173 (BatchN	(None, 8, 8, 192)	576	conv2d_173[0][0]
activation_173 (Activation)	(None, 8, 8, 192)	0	batch_normalization_173[0][0]

conv2d_174 (Conv2D) (None, 8, 8, 224) 129024 activation_173[0][0]

batch_normalization_174 (BatchN (None, 8, 8, 224) 672 conv2d_174[0][0]

activation_174 (Activation) (None, 8, 8, 224) 0 batch_normalization_174[0][0]

conv2d_172 (Conv2D) (None, 8, 8, 192) 399360 block8_2_ac[0][0]

conv2d_175 (Conv2D) (None, 8, 8, 256) 172032 activation_174[0][0]

batch_normalization_172 (BatchN (None, 8, 8, 192) 576 conv2d_172[0][0]

batch_normalization_175 (BatchN (None, 8, 8, 256) 768 conv2d_175[0][0]

activation_172 (Activation) (None, 8, 8, 192) 0 batch_normalization_172[0][0]

activation_175 (Activation) (None, 8, 8, 256) 0 batch_normalization_175[0][0]

block8_3_mixed (Concatenate) (None, 8, 8, 448) 0 activation_172[0][0]
activation_175[0][0]

block8_3_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_3_mixed[0][0]

block8_3 (Lambda) (None, 8, 8, 2080) 0 block8_2_ac[0][0]
block8_3_conv[0][0]

block8_3_ac (Activation) (None, 8, 8, 2080) 0 block8_3[0][0]

conv2d_177 (Conv2D) (None, 8, 8, 192) 399360 block8_3_ac[0][0]

batch_normalization_177 (BatchN (None, 8, 8, 192) 576 conv2d_177[0][0]

activation_177 (Activation) (None, 8, 8, 192) 0 batch_normalization_177[0][0]

conv2d_178 (Conv2D) (None, 8, 8, 224) 129024 activation_177[0][0]

batch_normalization_178 (BatchN (None, 8, 8, 224) 672 conv2d_178[0][0]

activation_178 (Activation) (None, 8, 8, 224) 0 batch_normalization_178[0][0]

conv2d_176 (Conv2D) (None, 8, 8, 192) 399360 block8_3_ac[0][0]

conv2d_179 (Conv2D) (None, 8, 8, 256) 172032 activation_178[0][0]

batch_normalization_176 (BatchN (None, 8, 8, 192) 576 conv2d_176[0][0]

batch_normalization_179 (BatchN (None, 8, 8, 256) 768 conv2d_179[0][0]

activation_176 (Activation) (None, 8, 8, 192) 0 batch_normalization_176[0][0]

activation_179 (Activation) (None, 8, 8, 256) 0 batch_normalization_179[0][0]

block8_4_mixed (Concatenate) (None, 8, 8, 448) 0 activation_176[0][0]
activation_179[0][0]

block8_4_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_4_mixed[0][0]

block8_4 (Lambda) (None, 8, 8, 2080) 0 block8_3_ac[0][0]
block8_4_conv[0][0]

block8_4_ac (Activation) (None, 8, 8, 2080) 0 block8_4[0][0]

conv2d_181 (Conv2D) (None, 8, 8, 192) 399360 block8_4_ac[0][0]

batch_normalization_181 (BatchN (None, 8, 8, 192) 576 conv2d_181[0][0]

activation_181 (Activation) (None, 8, 8, 192) 0 batch_normalization_181[0][0]

conv2d_182 (Conv2D) (None, 8, 8, 224) 129024 activation_181[0][0]

batch_normalization_182 (BatchN (None, 8, 8, 224) 672 conv2d_182[0][0]

activation_182 (Activation) (None, 8, 8, 224) 0 batch_normalization_182[0][0]

conv2d_180 (Conv2D) (None, 8, 8, 192) 399360 block8_4_ac[0][0]

conv2d_183 (Conv2D) (None, 8, 8, 256) 172032 activation_182[0][0]

batch_normalization_180 (BatchN (None, 8, 8, 192) 576 conv2d_180[0][0]

batch_normalization_183 (BatchN (None, 8, 8, 256) 768 conv2d_183[0][0]

activation_180 (Activation) (None, 8, 8, 192) 0 batch_normalization_180[0][0]

activation_183 (Activation) (None, 8, 8, 256) 0 batch_normalization_183[0][0]

block8_5_mixed (Concatenate) (None, 8, 8, 448) 0 activation_180[0][0]
activation_183[0][0]

block8_5_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_5_mixed[0][0]

block8_5 (Lambda) (None, 8, 8, 2080) 0 block8_4_ac[0][0]
block8_5_conv[0][0]

block8_5_ac (Activation) (None, 8, 8, 2080) 0 block8_5[0][0]

conv2d_185 (Conv2D) (None, 8, 8, 192) 399360 block8_5_ac[0][0]

batch_normalization_185 (BatchN (None, 8, 8, 192) 576 conv2d_185[0][0]

activation_185 (Activation) (None, 8, 8, 192) 0 batch_normalization_185[0][0]

conv2d_186 (Conv2D) (None, 8, 8, 224) 129024 activation_185[0][0]

batch_normalization_186 (BatchN (None, 8, 8, 224) 672 conv2d_186[0][0]

activation_186 (Activation) (None, 8, 8, 224) 0 batch_normalization_186[0][0]

conv2d_184 (Conv2D) (None, 8, 8, 192) 399360 block8_5_ac[0][0]

conv2d_187 (Conv2D) (None, 8, 8, 256) 172032 activation_186[0][0]

batch_normalization_184 (BatchN (None, 8, 8, 192) 576 conv2d_184[0][0]

batch_normalization_187 (BatchN (None, 8, 8, 256) 768 conv2d_187[0][0]

activation_184 (Activation) (None, 8, 8, 192) 0 batch_normalization_184[0][0]

activation_187 (Activation) (None, 8, 8, 256) 0 batch_normalization_187[0][0]

block8_6_mixed (Concatenate) (None, 8, 8, 448) 0 activation_184[0][0]
activation_187[0][0]

block8_6_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_6_mixed[0][0]

block8_6 (Lambda) (None, 8, 8, 2080) 0 block8_5_ac[0][0]
block8_6_conv[0][0]

block8_6_ac (Activation) (None, 8, 8, 2080) 0 block8_6[0][0]

conv2d_189 (Conv2D) (None, 8, 8, 192) 399360 block8_6_ac[0][0]

batch_normalization_189 (BatchN (None, 8, 8, 192) 576 conv2d_189[0][0]

activation_189 (Activation) (None, 8, 8, 192) 0 batch_normalization_189[0][0]

conv2d_190 (Conv2D) (None, 8, 8, 224) 129024 activation_189[0][0]

batch_normalization_190 (BatchN (None, 8, 8, 224) 672 conv2d_190[0][0]

activation_190 (Activation) (None, 8, 8, 224) 0 batch_normalization_190[0][0]

conv2d_188 (Conv2D) (None, 8, 8, 192) 399360 block8_6_ac[0][0]

conv2d_191 (Conv2D) (None, 8, 8, 256) 172032 activation_190[0][0]

batch_normalization_188 (BatchN (None, 8, 8, 192) 576 conv2d_188[0][0]

batch_normalization_191 (BatchN (None, 8, 8, 256) 768 conv2d_191[0][0]

activation_188 (Activation) (None, 8, 8, 192) 0 batch_normalization_188[0][0]

activation_191 (Activation) (None, 8, 8, 256) 0 batch_normalization_191[0][0]

block8_7_mixed (Concatenate) (None, 8, 8, 448) 0 activation_188[0][0]
activation_191[0][0]

block8_7_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_7_mixed[0][0]

block8_7 (Lambda) (None, 8, 8, 2080) 0 block8_6_ac[0][0]
block8_7_conv[0][0]

block8_7_ac (Activation) (None, 8, 8, 2080) 0 block8_7[0][0]

conv2d_193 (Conv2D) (None, 8, 8, 192) 399360 block8_7_ac[0][0]

batch_normalization_193 (BatchN (None, 8, 8, 192) 576 conv2d_193[0][0]

activation_193 (Activation) (None, 8, 8, 192) 0 batch_normalization_193[0][0]

conv2d_194 (Conv2D) (None, 8, 8, 224) 129024 activation_193[0][0]

batch_normalization_194 (BatchN (None, 8, 8, 224) 672 conv2d_194[0][0]

activation_194 (Activation) (None, 8, 8, 224) 0 batch_normalization_194[0][0]

conv2d_192 (Conv2D) (None, 8, 8, 192) 399360 block8_7_ac[0][0]

conv2d_195 (Conv2D) (None, 8, 8, 256) 172032 activation_194[0][0]

batch_normalization_192 (BatchN (None, 8, 8, 192) 576 conv2d_192[0][0]

batch_normalization_195 (BatchN (None, 8, 8, 256) 768 conv2d_195[0][0]

activation_192 (Activation) (None, 8, 8, 192) 0 batch_normalization_192[0][0]

activation_195 (Activation) (None, 8, 8, 256) 0 batch_normalization_195[0][0]

block8_8_mixed (Concatenate) (None, 8, 8, 448) 0 activation_192[0][0]
activation_195[0][0]

block8_8_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_8_mixed[0][0]

block8_8 (Lambda) (None, 8, 8, 2080) 0 block8_7_ac[0][0]
block8_8_conv[0][0]

block8_8_ac (Activation) (None, 8, 8, 2080) 0 block8_8[0][0]

conv2d_197 (Conv2D) (None, 8, 8, 192) 399360 block8_8_ac[0][0]

batch_normalization_197 (BatchN (None, 8, 8, 192) 576 conv2d_197[0][0]

activation_197 (Activation) (None, 8, 8, 192) 0 batch_normalization_197[0][0]

conv2d_198 (Conv2D) (None, 8, 8, 224) 129024 activation_197[0][0]

batch_normalization_198 (BatchN (None, 8, 8, 224) 672 conv2d_198[0][0]

activation_198 (Activation) (None, 8, 8, 224) 0 batch_normalization_198[0][0]

conv2d_196 (Conv2D) (None, 8, 8, 192) 399360 block8_8_ac[0][0]

conv2d_199 (Conv2D) (None, 8, 8, 256) 172032 activation_198[0][0]

batch_normalization_196 (BatchN (None, 8, 8, 192) 576 conv2d_196[0][0]

batch_normalization_199 (BatchN (None, 8, 8, 256) 768 conv2d_199[0][0]

activation_196 (Activation) (None, 8, 8, 192) 0 batch_normalization_196[0][0]

activation_199 (Activation) (None, 8, 8, 256) 0 batch_normalization_199[0][0]

block8_9_mixed (Concatenate) (None, 8, 8, 448) 0 activation_196[0][0]
activation_199[0][0]

block8_9_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_9_mixed[0][0]

block8_9 (Lambda) (None, 8, 8, 2080) 0 block8_8_ac[0][0]
block8_9_conv[0][0]

block8_9_ac (Activation) (None, 8, 8, 2080) 0 block8_9[0][0]

conv2d_201 (Conv2D) (None, 8, 8, 192) 399360 block8_9_ac[0][0]

batch_normalization_201 (BatchN (None, 8, 8, 192) 576 conv2d_201[0][0]

activation_201 (Activation) (None, 8, 8, 192) 0 batch_normalization_201[0][0]

conv2d_202 (Conv2D) (None, 8, 8, 224) 129024 activation_201[0][0]

batch_normalization_202 (BatchN (None, 8, 8, 224) 672 conv2d_202[0][0]

activation_202 (Activation) (None, 8, 8, 224) 0 batch_normalization_202[0][0]

conv2d_200 (Conv2D) (None, 8, 8, 192) 399360 block8_9_ac[0][0]

conv2d_203 (Conv2D) (None, 8, 8, 256) 172032 activation_202[0][0]

batch_normalization_200 (BatchN (None, 8, 8, 192) 576 conv2d_200[0][0]

batch_normalization_203 (BatchN (None, 8, 8, 256) 768 conv2d_203[0][0]

activation_200 (Activation) (None, 8, 8, 192) 0 batch_normalization_200[0][0]

activation_203 (Activation) (None, 8, 8, 256) 0 batch_normalization_203[0][0]

block8_10_mixed (Concatenate) (None, 8, 8, 448) 0 activation_200[0][0]
activation_203[0][0]

block8_10_conv (Conv2D) (None, 8, 8, 2080) 933920 block8_10_mixed[0][0]

block8_10 (Lambda) (None, 8, 8, 2080) 0 block8_9_ac[0][0]
block8_10_conv[0][0]

conv_7b (Conv2D) (None, 8, 8, 1536) 3194880 block8_10[0][0]

conv_7b_bn (BatchNormalization) (None, 8, 8, 1536) 4608 conv_7b[0][0]

conv_7b_ac (Activation) (None, 8, 8, 1536) 0 conv_7b_bn[0][0]

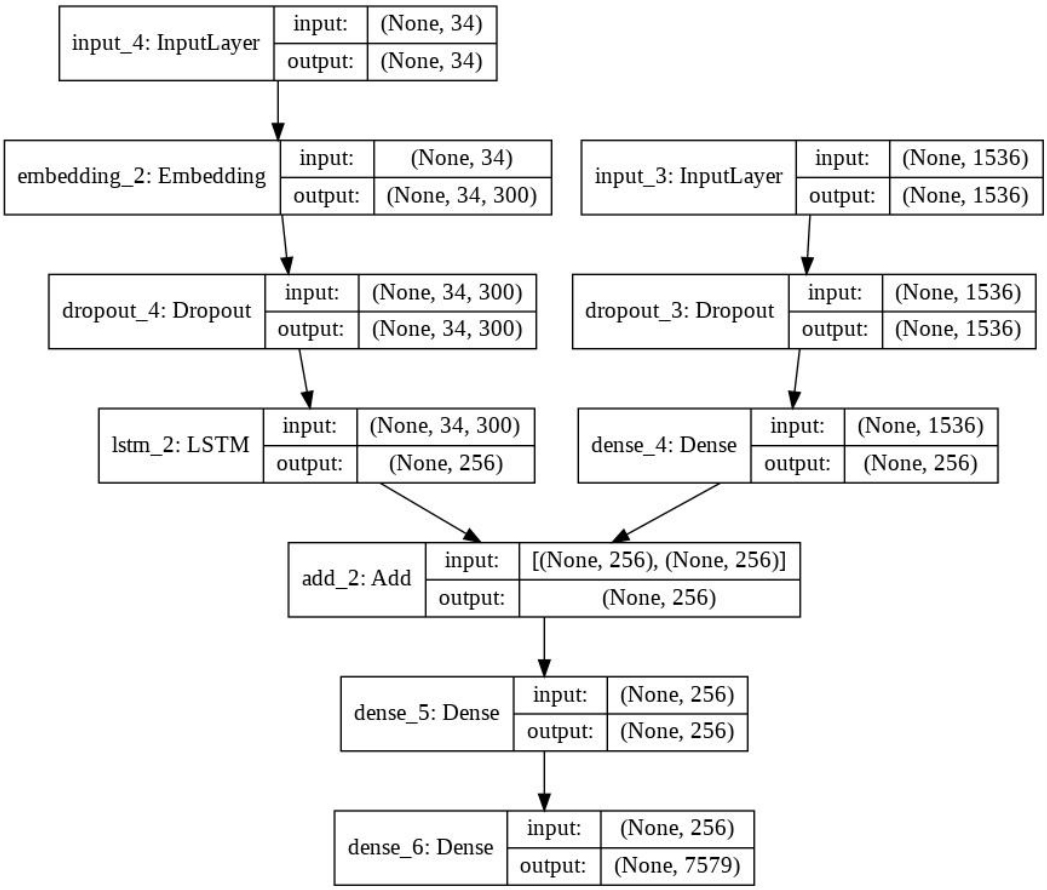
avg_pool (GlobalAveragePooling2 (None, 1536) 0 conv_7b_ac[0][0]

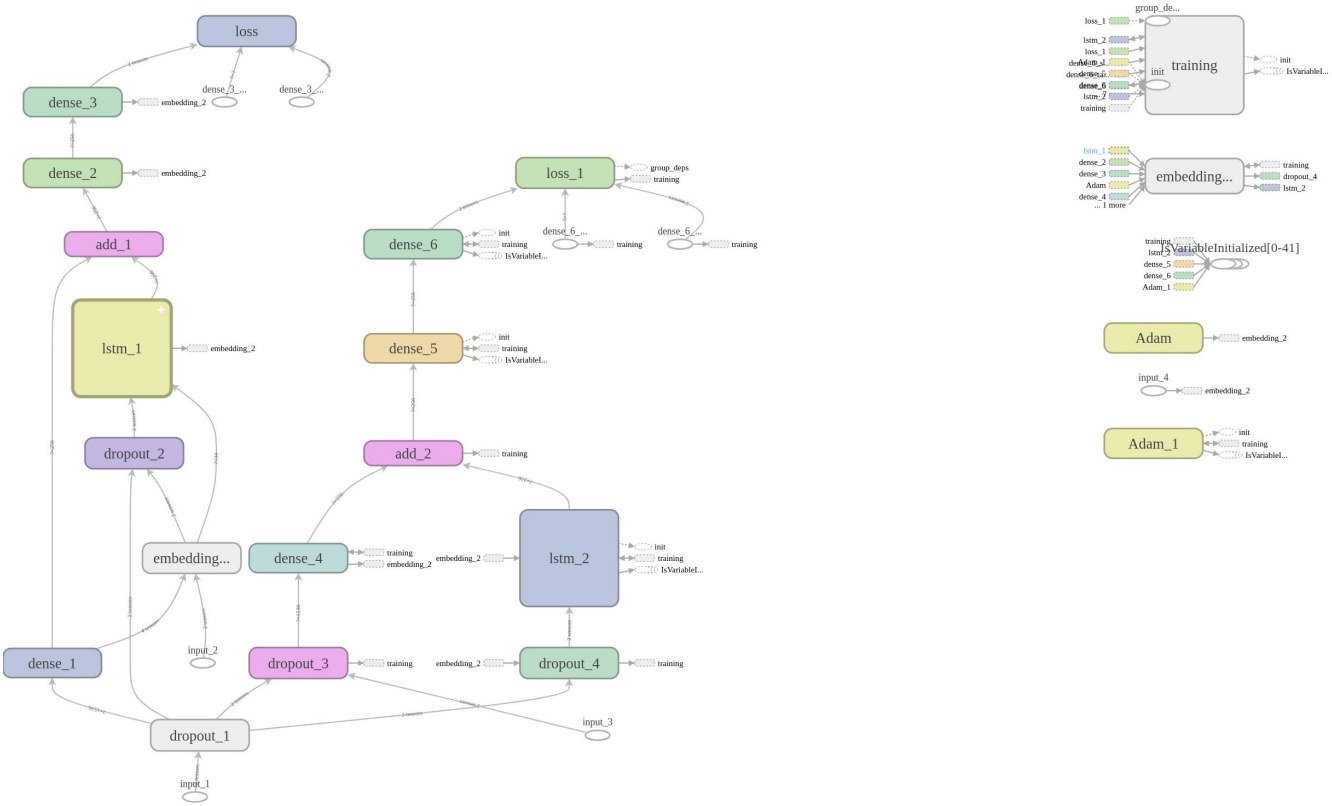
Total params: 54,336,736
Trainable params: 54,276,192
Non-trainable params: 60,544

Neural Network Architecture

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 34)	0	
input_1 (InputLayer)	(None, 1536)	0	
embedding_1 (Embedding)	(None, 34, 300)	2273700	input_2[0][0]
dropout_1 (Dropout)	(None, 1536)	0	input_1[0][0]
dropout_2 (Dropout)	(None, 34, 300)	0	embedding_1[0][0]
dense_1 (Dense)	(None, 256)	393472	dropout_1[0][0]
lstm_1 (LSTM)	(None, 256)	570368	dropout_2[0][0]
add_1 (Add)	(None, 256)	0	dense_1[0][0] lstm_1[0][0]
dense_2 (Dense)	(None, 256)	65792	add_1[0][0]
dense_3 (Dense)	(None, 7579)	1947803	dense_2[0][0]

Total params: 5,251,135
Trainable params: 5,251,135
Non-trainable params: 0





Tensorboard Model

Implementation

Flow of developing the caption generator:

1. Photo and Caption Dataset
2. Prepare Photo Data
3. Prepare Text Data
4. Develop Deep Learning Model
5. Train With Progressive Loading
6. Evaluate Model
7. Generate New Captions

For our model, we have made a tensorboard callback for monitoring losses.

We wanted to fit a model using a GPU, then we were not be able to fit the data into memory of an average GPU video card.

One solution is to progressively load the photos and descriptions as-needed by the model.

Keras supports progressively loaded datasets by using the *fit_generator()* function on the model. A generator is the term used to describe a function used to return batches of samples for the model to train on. This can be as simple as a standalone function, the name of which is passed to the *fit_generator()* function when fitting the model.

As a reminder, a model is fit for multiple epochs, where one epoch is one pass through the entire training dataset, such as all photos. One epoch is comprised of multiple batches of examples where the model weights are updated at the end of each batch.

Result and Evaluation Summary

We have used Inception ResNet V2 and Fasttext word embeddings to improve the accuracy. BLEU score was used as a metric with following results on sub-datasets.

VGG 16 gives us the following result on validation dataset.

BLEU -1 : 0.505694
BLEU -2 : 0.257299
BLEU -3 : 0.173234
BLEU -4 : 0.077892

Inception ResNet V2 with Fasttext wiki pre-trained embeddings gives us the following results:

BLEU - 1 : 0.539671
BLEU - 2 : 0.293452
BLEU - 3 : 0.203698
BLEU - 4 : 0.096276

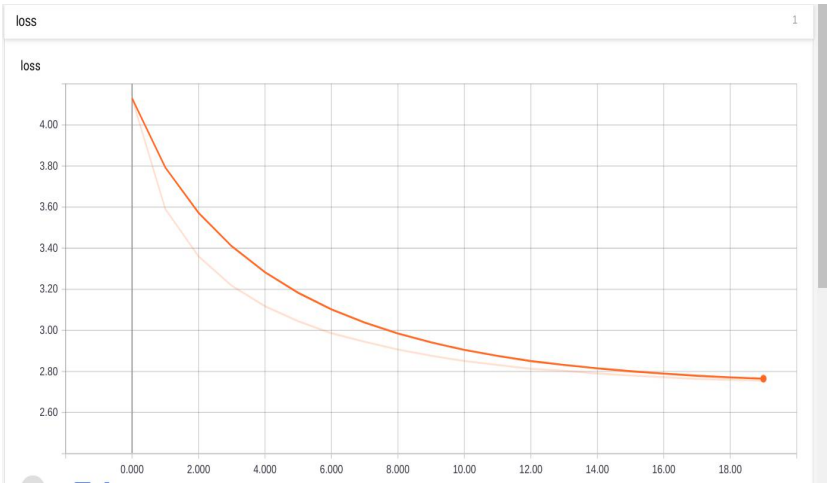
Validation ^

BLEU - 1 : 0.548384
BLEU - 2 : 0.306833
BLEU - 3 : 0.213625
BLEU - 4 : 0.103650

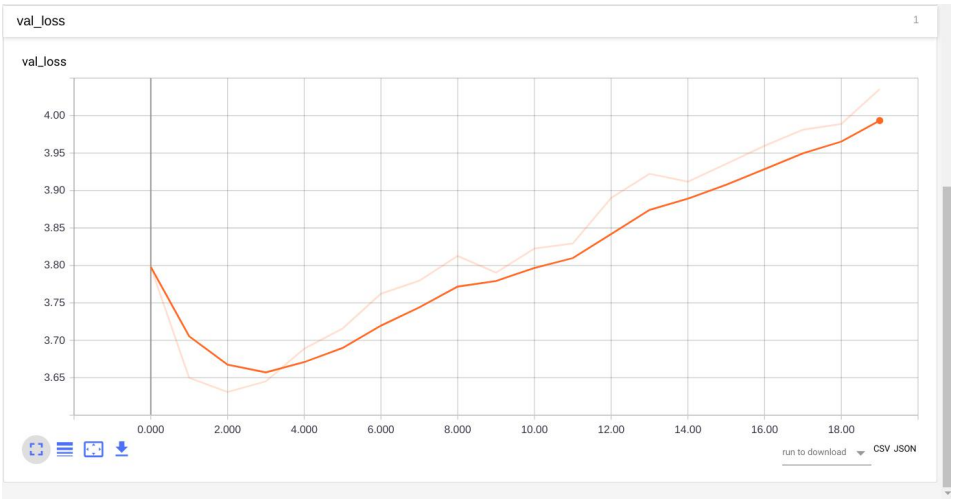
Test ^

BLEU - 1 : 0.547827
BLEU - 2 : 0.307524
BLEU - 3 : 0.215886
BLEU - 4 : 0.104762

Train^



Training Loss



Validation Loss