Deep Art: Learning Artistic Style via Residual and Capsule Networks

Pierre Eugene Valassakis (ucabpev, 1010134)
Julia Gomes (ucabjmg, 17107203)
Liam Eloie (zcaplel, 13019996)
Vasileios Papastefanopoulos (ucabvp1, 17046496)

Abstract

We experiment with various architectures in an effort to build an artistic style classifier capable of classifying artwork from the WikiArt dataset according to seven different art genres, such as cubism and impressionism. To study how object identification tasks relate to style recognition tasks, we experiment with transfer learning from the ResNet architecture trained on the ImageNet dataset and visualize the activations of the final layers. We additionally test the CapsNet model on our data, an architecture which shows great improvements on MNIST but has not yet been used in other domains. Finally, we build a custom model that merges the ResNet and CapsNet architectures.

1. Introduction

Convolutional Neural Networks (CNN) are the foundational framework for most computer vision tasks today. CNNs have been very successful in tasks such as object recognition, with many modern architectures able to outperform humans in image classification challenges on ImageNet and other datasets. However, much less research has been applied to classification tasks pertaining to artwork. We believe that optimizing neural networks to learn artistic style would have widespread benefits for both artists and collectors alike.

WikiArt (WikiArt; cs chan) is one of the first and most comprehensive online art encyclopedias, developed in 2013. The WikiArt dataset currently has 150,000 artworks spanning 54 genres and is continually adding to their database. Recently, the company Artsy has built on the WikiArt dataset and collected 800,000 art pieces to use for their Art Genome Project. Artsy is currently hand-labelling each piece of artwork with 1,000 genes in order to match art collectors with new pieces based on their preferences. Needless to say, hand-labelling images is a time-consuming and costly process.

Our goal in this project is to 1) build an art classifier that can automatically label the genre of new pieces of art, and 2) investigate how architectures designed for object identification transfer over to the domain of artistic style classification. We will focus on transfer learning with the ResNet-50 architecture (He et al., 2015) as well as the recently developed CapsNet architecture (Sabour et al., 2017), which has shown great improvements on digit recognition for the MNIST dataset but has not yet been tested in other domains. In addition, we will implement our own architecture that combines features from both networks. Finally, we will visualize the activations of our networks in order to see which features our model notices.

2. Background

2.1. Related work

Art classification is a complex task, often a challenge, even for experts. A variety of different methods have been employed recently for art classification including ultraviolet fluorescence, x-radiography and paint sampling (Johnson et al., 2008). Previous work on art classification includes both traditional methods, based on hand-crafted features as well as methods where the features are automatically learned from the images. While feature engineering can be useful in terms of understanding the behavior of the classifier and interpreting the results, classifiers based on handcrafted features do not in most cases perform well on art classification. Learning the features automatically from the images, usually leads to classifiers of higher accuracy, as shown in (Tan et al., 2016), where an accuracy of 54.5% over 25 styles was achieved.

In (Sablatnig et al., 1998), the structure of brush strokes was examined as a main attribute to identify personal artistic styles. A computer-aided top-down classification and recognition system was developed, proving that stroke arrangements can be a stable method to detect artist-specific styles. (Lombardi), took a more general, style-independent approach on painting classification. More specifically, the palette description algorithm was used to compare color content in images and performed similarly to color description techniques (e.g. hue histogram, saturation histogram) requiring less storage. Results showed that algorithms preserving frequency and spatial color features do not necessarily optimize the accuracy of painting classification. Supervised (k-NN) and unsupervised techniques (Agglomerative Hierarchical Clustering, Self-Organizing Maps, Multidimensional Scaling) were demonstrated to be equivalent to data quality metrics. Previous work (Jou & Agrawal, 2012) also includes training on basic (color histogram) and advanced features (HOG descriptors) to classify paintings by 5 artists. Several classification techniques were used, with the Nave Bayes classifier giving the maximum accuracy of 65%. In a more recent approach (Mensink & Van Gemert, 2014), image features were encoded using Fisher vectors to classify digitized artworks (paintings, photographs etc.) of the Rijksmuseum dataset by artist, type, materials and year of creation. 1-vs-all linear SVM classifiers were used for all creators in artist classification, type categorization and material labeling. For estimating the year of creation, the use of an SVM classifier followed by a type regressor was proven to be more effective than a single regressor.

Regarding the use of automatic feature learning, a Convolutional Neural Network (PigeoNET) was trained in (Van Noord et al., 2015) to recognize and attribute unseen artworks to artists. The Rijksmuseum Challenge dataset was used to optimize and assess the performance of PigeoNET. Results showed that PigeoNET is very accurate in attributing artworks (more than 70%) and will probably perform even better in more extended datasets. In another prior work (Bar et al.), a CNN was applied to extract low-level visual features from the ImageNet dataset. The deep network was combined with the PiCoDes binary descriptor and different low-level descriptors to recognize artistic styles. The results showed that the most accurate method was the combination of Decaf encoding with PiCoDes, which outperformed low-level descriptors. A more recent approach (Lecoutre et al., 2017), also employed deep architectures: Deep neural networks were pre-trained on ImageNet to detect artistic styles. It was proven that retraining the layers of the network improves the performance (20 retrained layers gave the optimal performance) of the method. The model was tested on a Wikipaintings dataset and an independent source, showing that the classifier did not overfit. ResNet architecture performed better than AlexNet reaching over 90% accuracy.

2.2. Residual Networks

When building neural networks, we prefer to add depth over width in order to minimize parameters and prevent over-fitting. In theory, a neural network can learn any representation with a sufficiently deep architecture. However, deeper networks are prone to saturation and vanishing gradient problems, causing the error rate to plateau early on. In practice, the training error rate in deeper networks is often higher than the training error rate in shallower networks.

The ResNet solves this problem by introducing an identity short-cut connection, illustrated below. It is easier for these residual blocks to learn the identity mapping, in which case the additional layers do not modify the input. Because the ResNet can use identity blocks to imitate a shallower model, adding layers to the ResNet will not degrade performance. In fact, experiments show that a 1001-layer deep ResNet outperforms shallower versions. (Kaiming He) To summarize, residual networks are deeper than vanilla neural networks but require a similar number of weights and do not have problems with saturation.

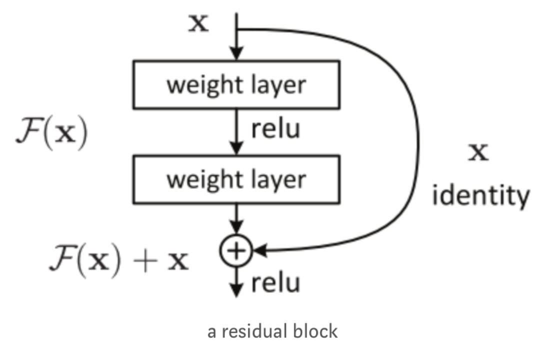


Figure 1. An illustration of the residual block (Fung)

2.3. Transfer Learning

Transfer learning describes the process of training a model on a new task by transferring and utilizing the knowledge learned by another model that was trained on a related task (Torrey & Shavlik, 2009). A more formal definition of transfer learning is given in (Pan & Yang, 2010) by Pan and Yang:

"Given a source domain D_S and learning task T_S, a target domain D_T and learning task T_T, transfer learning aims to help improve the learning of the target predictive function $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S, where $D_S \neq D_T$, or $T_S \neq T_T$."

In practice, transfer learning has been applied with success in Machine Vision, Reinforcement Learning and Natural Language Processing. Regarding Machine Vision, (Sharif Razavian et al., 2014) trained a new model using features of a large model, built on ImageNet. In (Ganin

& Lempitsky, 2015) a model capable of confusing source and target domains was created, using transfer learning. Furthermore, both (Bousmalis et al., 2016) and (Sener et al., 2016) employed transfer learning techniques to train models on domain-invariant representations. In reinforcement learning, progressive neural networks that can learn from already trained models have been discussed in (Rusu et al., 2016), whereas in (Fernando et al., 2017) successful transfer learning is demonstrated in Pathnet, a neural network that uses genetic algorithms during training. In natural language processing, transfer learning has been applied to text classification (Do & Ng, 2006), (Raina et al., 2006) as well as spam filtering (Bickel). Other novel approaches involving transfer learning include (Ruder et al., 2017), where transfer learning is employed to extend Knowledge Distilling (proposed by Hinton et al in (Hinton et al., 2015)) to the domain adaptation and (Ruder & Plank, 2017), where Bayesian optimization was applied for transfer learning data selection.

2.4. Capsule Networks

A major limitation of Convolutional Neural Networks is that they are unable to capture the relative spatial relationship between features in an image and are not viewpoint invariant. To be concrete, this means that it is difficult for a CNN to identify a nose when it is looking at it from a new angle and the CNN will also not necessarily notice if the nose is in the wrong position on a face. (Colyer) In an attempt to address this inefficiency, another type of neural network was recently introduced by Sabour et al. (Sabour et al., 2017): The Capsule Network. Capsule networks use capsules to retain viewpoint invariant knowledge and spatial structure. These capsules perform complex internal computations on their inputs and then output a vector which encodes a variety of instantiation parameters related to a particular feature, such as position, orientation, scale, thickness, and texture. The length of the capsule output vector is squashed to be no more than one so that it can represent the probability that a feature is present, while the elements of the vector encode its properties. Thus when an entity is viewed in a new orientation, the length of the associated vector (probability it exists) should stay the same while the orientation (instantiation parameters) will adjust accordingly. (Pechyonkin)

For instance, when working with the MNIST dataset, each capsule holds various spatial information about each digit. Figure 2 illustrates how spatial information about the digits is stored inside the Capsule Network for MNIST.

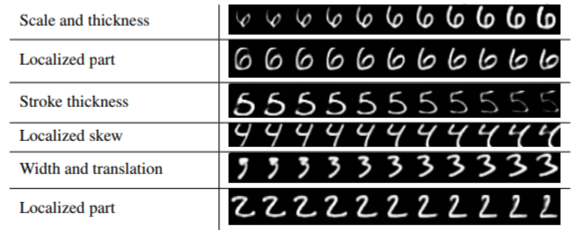


Figure 2. How spatial information about the MNIST digits stored inside the Capsule Network (Sabour et al., 2017)

The network then uses a dynamic routing system called routing-by-agreement to send the vector to the appropriate parent in the next layer. For example, a capsule which observes noses will send its vector to the parent capsule looking for faces. This serves as an alternative to max-pooling, the commonly used routing system which causes neurons to ignore all but the most active feature in the previous layer. Finally, the CapsNet uses a margin loss function at the last digit class layer, where a long output vector in digit class k implies that class k is present. The total loss is the sum of the losses over all digit capsules, adding a penalty if 1) the class is present and the length of the associated vector is small, or 2) the class is absent and the length of the relevant vector is large. In the original paper, the formula is:

$$L_k = T_k \max(0, m^+ - \|v\|^2) + \lambda(1 - T_k)\max(0, \|v_k\| - m^-)^2$$

where $m^+ = 0.9$, $m^- = 0.1$, and $T_k = 1$ iff a digit from class k is present. (Sabour et al., 2017)

A simple Capsule Network architecture is shown in Figure 1.

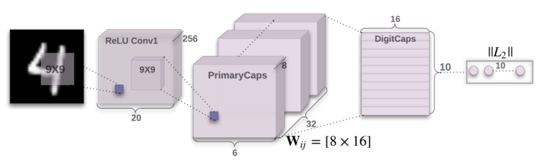


Figure 3. A simple Capsule Network Architecture (Sabour et al., 2017)

Experiments demonstrate that Capsule Networks encode more robust representations for digit classes on MNIST than traditional convolutional networks: Result show that an under-trained Capsule Network, which achieved 99.23% accuracy on the expanded MNIST test set achieved 79% accuracy on the affnist test set, outperforming a conventional convolutional model with a comparable number of parameters, as the latter scored similar accuracy (99.22%) on the expanded MNIST test set but only managed 66% on the affnist test set (Sabour et al., 2017).

2.5. Visualisation

Each convolutional layer learns template matching filters after training. These filters have the highest output when their input image is close to the template. For example, a filter might have a nose template that activates when there is a nose in the input image. Activation maximization is a method used to visualize such a template. If we use activation maximization to visualize the nose filter, for instance, we should produce an image that looks somewhat like a nose.

In order to generate an input image that maximizes the filter output activations, the Keras $visualize\_activation$ function takes raw image pixels as input then tries to minimize the Activation Maximization Loss, because a small activation loss implies a large filter activation. Thus we have a simple optimization problem - we minimize the Activation Maximization Loss by taking the derivative of the loss with respect to the input, then use gradient descent to update the input until we have generated the optimal template. This final template is the image which $visualize\_activation$ returns. To implement the function in Keras, we simply specify the model and the arguments for $filter\_indices$ and $layer\_idx$, directing the function to the appropriate filter in the layer of interest. For example, we could direct the function to visualize a filter in the output layer, specifically the filter that classifies an image as a cat. There are also optional parameters to modify the back-propagation technique, the gradient descent method, and other arguments, as explained in the documentation. (Kotikalapudi)

3. Dataset

3.1. Overview

In deep learning systems it has been well established that both the architecture and the amount of training data used are fundamentally important to the success of a model. It is often said that, 'a dumb algorithm with lots and lots of data beats a clever one with modest amounts of it' (Domingos, 2012). Motivated by this idea, a large dataset known as the 'WikiArt Dataset' was chosen to train the deep neural models (WikiArt; cs chan). WikiArt is one of the largest publicly known art datasets consisting of around 100,000 paintings labelled by their artist, genre, and style, which is particularly suitable for supervised learning. For the purposes of this paper, the artist and genre will be discarded, allowing supervised learning to be performed on the style of the paintings alone.

The paintings found in the WikiArt dataset come from 26 different styles, spanning from Realism to Abstract Expressionism, as well as varying widely in terms of dimension. As is often the case with real world datasets, there is class imbalance amongst the styles and as such needs to be dealt with accordingly.

Style Labels	
Class ID	**Type**
0	Art Nouveau Modern
1	Baroque
2	Cubism
3	Expressionism
4	Impressionism
5	Symbolism
6	Realism

Table 1. The different types of art styles and their corresponding class ID present in the reduced data set.

3.2. Preprocessing

Due to limited computational resources, it was necessary to reduce the number of data used to train the models. This was done by (1) Reducing the number of classes to be classified from 26 to 7, and (2) Choosing randomly 2000 paintings from each class, discarding the rest. Not only this resulted in a manageable dataset within resources, but also tackled the problem of imbalanced classes. The remaining style types and their corresponding class ID in our reduced dataset can be seen in **??**. To test the generality of our models, we split this reduced data set into a training and a test set with a ratio of 80:20. Lastly, in order for the models to take these paintings as input, they must have the same dimension. There are two main ways of doing this: (1) scale all paintings to the same size, or (2) take a number of crops with specified dimensions. As we are only concerned with the style of the paintings, it is intuitively more important to preserved pixel-level details rather than the larger structures of a painting. These details are not preserved by scaling paintings and as such this problem was solved by taking crops. An important consideration to make during the cropping process is the dimension of the smallest painting in the dataset. This dimension acts as an upper-bound for the cropping dimension and is crucial for determining how much larger structure is to be considered within the model. To avoid the case where the cropping dimension is too small, leading to loss of important information, the smallest 10% of paintings were removed. The reduced dataset now consists of 9000 paintings, where each painting is split into four crops of our chosen dimensions.

Given the large amount of data after cropping and the processing needed before passing the data into the models, a decision was made to store the cropped images to disk. This decision was made due to limited amount of memory allocated, allowing for more data to be used to train the models as well as a larger crop to be taken, hopefully giving rise to an improvement in accuracy.

4. Methodology

4.1. Setup

As mentioned previously, our purpose is to compare performances of different networks distinguishing between the 7 different styles in our dataset. In order to do so we did the following:

Transfer Learning on the ResNet.

First, we experimented with transfer learning on the ResNet-50 network. Specifically, we used the pre-trained weights on the ImageNet dataset, and then fine-tuned them using our paintings dataset. It is important to note that during the procedure, it was necessary to replace the last 1000-class softmax layer of the network with a 7 class softmax layer, which corresponded to our seven classes of styles. We additionally experimented with various fine-tuning methods, described below:

- In Experiment 1, we fine-tuned all layers of the ResNet using our dataset.
- In Experiment 2, we froze the all the layers up to and including "activation_13", about $\frac{1}{3}$ of the network which had a total of 49 activation layers (c.f. appendix A). When fine-tuning the weights on our dataset, the frozen layers do not learn. The rationale behind such a setup is that we expect the initial layers on a CNN to merely detect edges and simple shapes, so those initial layers do not need to be fine-tuned for a new dataset. Training those layers from scratch would only result in longer training times, so it is a good idea to experiment with leaving those layers untrained.
- In Experiment 3, we only fine-tune the final classification layer of the network. This indicates whether the features learned on the ImageNet dataset are also capable of discriminating between our paintings, in which case freezing all but the final layer should give good results.

CapsuleNet training from scratch

When we experimented with the CapsuleNet, we trained the model from scratch with a random initialisation. We did not use transfer learning because the CapsNet is fairly new and had only been trained on MNIST, which is very different from our dataset and has a resolution that is far too small for our purposes (28 by 28). In terms of computational complexity, the CapsNet took too long to train for us to experiment extensively with many alternatives and different methods of training. This is because the CapsNet has more elements to train, as it relies on activation vectors which encode both the probability of the presence of a feature (indicated by vector length) and the orientation of the feature (pose parameters). Specifically, according to Geoffrey Hinton, the transformation matrices have n^2 parameters rather than just n.

A custom Network trained from scratch

We noticed that the first block of the CapsuleNet was composed of a regular CNN, and hence thought it would be interesting to replace this by the ResNet network. The rationale behind this is that it might be more effective to use the representations of the ResNet (over the massive imageNet dataset), rather than learning representations from scratch through a standard convolutional layer (and our much smaller training set). As such, we created a custom network, which we named CustomNet, in the following way:

- We first used the convolutions of the ResNet (stripping away the ResNet output) and loaded the ImageNet weights.
- We then froze this part of the network so that these weights would be unaffected by training.
- We finally routed the output of this configuration through the capsule layers of the CapsNet, which produced the final seven class outputs.

In this way, we effectively replaced the convolutional layers in the CapsNet by a whole ResNet network. We also made an alternative model in which we added dropout layers to avoid over-fitting.

4.2. Evaluation metrics

In order to evaluate our models, gain an understanding of their behaviour and allow us to compare model performance, we relied on two different evaluation metrics:

Loss and Accuracy on the training and test sets

The primary performance measure we used for our networks is the classification accuracy. This is calculated as $\frac{n_{tp}}{n_{total}}$, the ratio between the number of correct classifications to the total number of examples, and is sensible in our case as the datasets we are using are very balanced. Along with the accuracy, we also keep track of the cost, which is the categorical cross-entropy for the ResNet and the margin loss for both the CapsNet and CustomNet. These metrics where evaluated on both the training and test sets.

Moreover, in order to gain insight into (1) how various hyper-parameters affected the performance of our networks, and (2) the effectiveness of training, we kept track of these metrics after each epoch of training.

Layer Visualisation
Second, we used layer visualizations to gain insight into the behaviour of our networks. More specifically, we applied the visualisation techniques described in section 2.5 to visualise the activations for each output class. This allowed us to visualize the input that would maximise the activation for each of our chosen styles. The reason we chose this visualisation technique over others is that it works well on all of our chosen architectures, and hence is well-suited for comparison purposes.

4.3. Implementation details

4.3.1. Code Structure, Libraries and Hardware

Our code base is composed by (1) a set of APIs implementing the helper functions we need in order to train, run and evaluate our models and (2) a set of scripts actually performing these functions. Each of the APIs is packaged in a python package in a folder named after its functionality. All in all, our project directory has the following structure:

```
DeepArt
|
 —— Preprocessing : Contains the functions
    necessary for the preprocessing
|   of the Data (extracting, cropping,
    splitting ect.).
|
 —— Capsnet : Contains the code for the
    capsuleNet model, which was
|   adapted from Xifeng Guo's
    implementation. [*]
|
 —— CustomNet : Contains the code the for
    CustomNet model, which was constructed
|   through  a combination of the Resnet
    and the Capsnet.
|
 —— TransferLearning : Contains the
    functions necessary for the transfer
|   learning on the Resnet.
|
 —— Visualisation : Contains the functions
    necessary for visualising the
|   layers and the training history of the
    models.
|
 —— Utilities : Contains general utilities
    helper functions.
|
 —— Scripts : Contains all the scripts that
     use the APIs above in order to
|    train, run, test and evaluate our
     models.
|
 —— SavedData : Folder containing all the
    saved data from the experiments
|   and trained models.
|
 —— Wikiart: The Dataset.
```

Listing 1. : (Guo)]DeepArt project code structure. [*]: (Guo)

In making our implementation, the main libraries we relied on are the following :

- **Keras.io** (Chollet et al., 2015), a deep learning API implemented on Tensorflow (Abadi et al., 2015) that has the advantage of higher level of abstraction compared to the later.
- **Keras-vis** (Kotikalapudi & contributors, 2017), a Layer-visualisation library.
- **XifengGuo/CapsNet-Keras** (Guo), a keras implementation of the capsuleNet network, which we adapted to our setup.

Finally, in order to run our scripts, we used UCL's *Blaze* server, a shared machine with four GeForce GTX titan X GPUs, which we were able to utilize thanks to Tensorflow (Abadi et al., 2015).

4.3.2. Main scripts

ResNet Script
In this script we are doing the transfer learning on the ResNet-50 network, according to the four experiments described in section 4.1. The paintings are loaded, split into training and test sets and cropped in order to conform to the ResNet input dimensions (and for all the input data to have the same size). Specifically, we used a pixel size of (224,224) pixels per crop, and performed 4 crops per painting. Then we ran the experiments described in section 4.1. Each experiment includes the following stages:

- Load the ResNet with the pre-trained ImageNet weights. To accomplish this, we use the ResNet-50 implementation in Keras.
- Change the last layer of the model to have 7 output classes.
- Freeze the number of layers appropriate for each experiment so that the weights of the frozen layers will not be fine-tuned.
- Fine-tune the model using our dataset.
- Save the re-trained model and training histories for later assessment.

To fine-tune the ResNet, we trained over 40 epochs, applying Adam optimization with a batch-size of 128 and categorical cross-entropy loss. We additionally use early-stopping on the validation set with a patience of 2, which which we chose as to save up computational resources, and as transfer-learning experiments usually require fewer epochs to fine-tune their weights. As we noted in the previous section, we track the loss and accuracy of the model throughout the procedure.

CapsNet Script
This script trains the CapsuleNet from scratch. Again, the paintings are loaded, cropped, then split into training and validation sets. Because of memory constraints on the *Blaze* machine, we applied a cropping dimension of (100,100) and used 4 crops per painting. We then trained the model over 10 epochs. The optimiser used was still Adam with a batch size of 16 (due to the limited memory resources). We used the margin loss function, described in section 2.4. At the end of training, the model is saved along with the training history and performance metrics at each epoch.

CustomNet Script
In the CustomNetwork, we used the output of the ResNet as the input to the capsule layers. In order to do so and respect consistency in layer-by-layer dimensionality, we needed to modify the kernel size of the capsule layer from 9 (value used in the CapsNet) to 3. We also added a dropout layer with dropout probability 0.5.

In the CustomNet training script, we load and crop the data in the same way, then train the CustomNet using input dimension (224,224), 10 epochs, batch size 32, the Adam optimiser, and the margin loss function. We also train the model with and without the dropout layers. Finally, we save the trained networks and the training history.

Visualisation Scripts
To visualise the network layers, we load the saved models, run the visualisation functionalities using the keras-vis API (Kotikalapudi & contributors, 2017), then save the visualisations in Portable Network Graphics (.png) format. We save one visualisation per output class, which as we saw in section 2.5 represents the input to the network that maximises the activation of that class.

5. Results and Discussion

5.1. ResNet Transfer Learning Results

The results from our transfer learning experiments with the ResNet are shown in Table 2. As we can see from the table, retraining all parameters of the ResNet yielded an accuracy of $\approx 0.43\%$ on the test set, freezing the first few layers gave $\approx 48\%$, and retraining only the last layer resulted in $\approx 52\%$ accuracy.

ResNet Experiments		
Frozen layers	**Train Accuracy**	**Test Accuracy**
None	0.6520	0.4340
Up to activation_13	0.7410	0.4805
All except last	0.6200	0.5260

Table 2. Testing and training accuracy using transfer learning with ResNet on the WikiArt dataset.

Table 2 clearly indicates that the ResNet model trained on the ImageNet dataset is better at discriminating between our classes. In fact, the highest score is obtained while only fine-tuning the output layer of the network, leaving the pre-trained weights unchanged. Possible justifications for this behaviour are that our paintings dataset is not large or varied enough, or that we are still under-fitting the data when we train from scratch and that the model would benefit from longer training time. It is also worth noting that the model had a higher training accuracy when we re-trained the layers after activation 13 (leaving approximately $\frac{2}{3}$ of the network to re-train) than when we re-trained only the last layer, even though the testing accuracy was lower. This discrepancy indicates that we may have over-fit the model.

In order to further understand our results, we can examine the graphs of the evolution of the test set accuracy over each epoch of training. These are shown in Fig. 4.

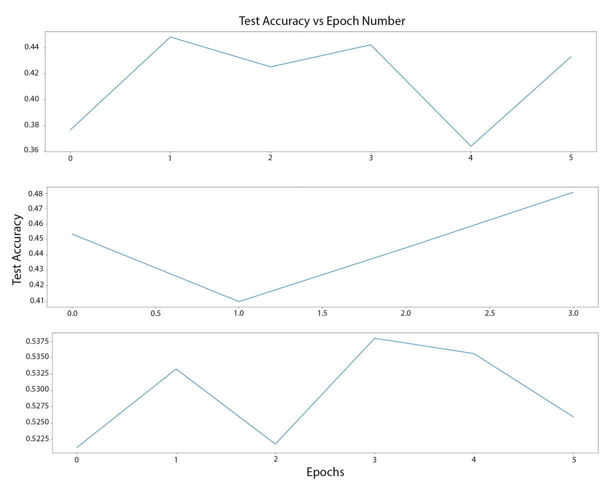


Figure 4. From top to bottom, these are the test accuracies of: Experiment 1, Experiment 2, Experiment 3

From looking at Figure 4, it is not clear that our networks had finished learning when we stopped training: Particularly in experiment 2, training was cut-off while displaying a clear upward trend in test set accuracy. This seems to indicate that a higher value for the patience of our networks should have been used, although we were restricted by the computational resources, as we could not monopolize the Blaze machine which is shared by all students. Nevertheless, experiments 1 and 3 display a test-set accuracy that seems to fluctuate randomly, so it is entirely possible that the results obtained are also a consequence of insufficient training data.

Finally, it is interesting to observe the visualization of the last layer activations for each of our style categories (c.f. section 3). The most illustrative example is the one of Experiment 2, which is shown in Fig. 5 below. It is clear that our network learned to recognize some of the styles better than others. For instance, in class 2 we can see sharp, geometrical edges cutting off each other at various angles creating an mesh of inter-tangled geometrical shapes. This is quite reminiscent of the style of Cubism which it represents!

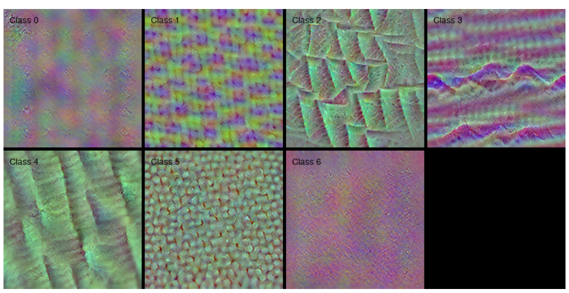


Figure 5. Visualisations of the activations in the last layer of the ResNet model from Experiment 2.

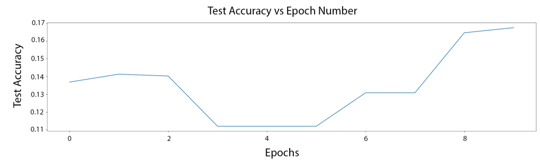
5.2. CapsNet Results

The results from the CapsNet training experiment are shown in Table 3. As we can see, the CapsNet obtained an accuracy of $\approx 16\%$, which is an indication that it has not learned much from training. This is not surprising, since the ResNet trained from scratch also had quite poor accuracy and indicates that we may have needed more training time and data.

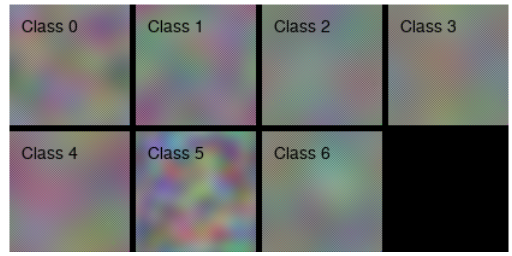
CapsNet Experiments	
Train Accuracy	**Test Accuracy**
0.1520	0.1680

Table 3. Performance of the CapsNet on the Wikiart dataset

As before, in order to draw further insights on why these results were obtained, we look at the test accuracy evolution during training and the visualisations of the last layer of the CapsNet. These are displayed in Fig. 6 below.



(a) Evolution of the test set accuracy of the Capsnet over each training epoch.



(b) Visualisation of the Capsnet dense layer.

Figure 6. Capsnet Evaluation Figures

From Fig.6 two issues become apparent that possibly prevented our network to learn:

1. We stopped training while the test set accuracy was on an upward trend. This seems to indicate that we are still in the under-determined case and that given more epochs to train, the CapsNet would have obtained a higher accuracy.

2. The visualisations are very blurry and pixelated. This indicates that the cropping size of (100,10) is not sufficient to correctly capture stylistic information in the paintings (eg, topological relationships).

In both cases it seems that in order to resolve these issues we would require further computational resources. The CapsNet is very slow to train and has an extremely large number of parameters, requiring both extensive training time and additional memory.

6. CustomNet Results

The results from training our CustomNet are shown in Table 4 below. As we can see, we achieve a test set accuracy of $\approx 52\%$ in both variants of the model. The model is clearly over-fitting to the training data, as the training accuracy is $\approx 98\%$ and $\approx 91\%$, respectively, which is nearly double the testing accuracy. The test accuracy on CustomNet is not higher than the test accuracy for the ResNet model in which we trained all except the last layer. This suggests that concatenating the capsule layers to the ResNet output does not improve performance. In Fig. 7, we plot the test set accuracy evolution over the epochs of training. It is apparent that, contrary to the ResNet and CapsNet, training seems to saturate after 2 epochs in both variants. Indeed, in both cases, the test set accuracy kept falling pretty regularly after the second epoch.

CustomNet Experiments		
	Train Accuracy	**Test Accuracy**
Without Dropout	0.9810	0.5150
With Dropout	0.9100	0.5175

Table 4. Performance of the CustomNet on the Wikiart dataset

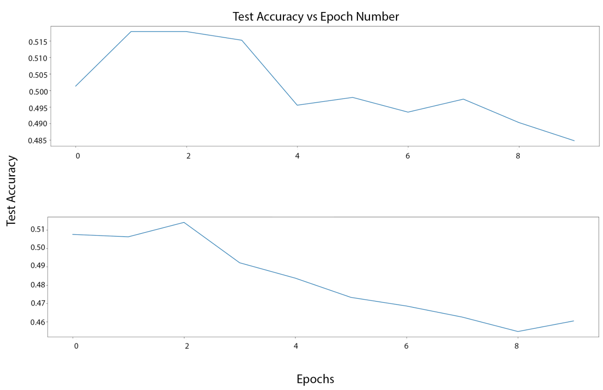


Figure 7. Evolution of the test set accuracy of the CustomNet over each training epoch.Top: Dropout case. Bottom: No-dropout case.

7. Conclusions

In conclusion, we found that is possible to train deep neural models able to classify artworks by their style. By experimenting with three different network architectures, we showed that applying transfer learning on the ResNet architecture yielded the best results. More specifically, we examined the effect of freezing different number of layers and discovered it is best to retrain only the final layer, whilst keeping the rest frozen. This illustrates the similarities between low-level features as well as the differences in high-level features between object-recognition and art style classification. Visualisations of final layer activation of the networks were generated, showing that these types of models are successfully able to detect specific details belonging to different art styles.

The second network architecture we experimented with was the recently developed capsule network. Unfortunately, the capsule network produced poor results when classifying artistic style. Possibly due to the small amount of training data, the model was unable to capture the trends that each style imposes and therefore failed to learn effective parameters. This is evident by inspection of the corresponding visualisations which appear to be random noise.

The last network architecture that we trained was a custom network in which we appended capsule output layers to the transfer-learned ResNet architecture. This achieved much better results that the capsule network, but did not perform better than the original ResNet model with transfer learning. This indicates that the capsule network was unable to learn low-level features of the art work, despite being able to perform extremely well on the MNIST dataset. However, we acknowledge that this may be due to insubstantial training time and data.

In the future, we would like to test the capsule network and custom network on a larger dataset and allow for longer training time. We are curious to see how the model would compare to the ResNet architecture without the computational resource constraints. If the same results hold for this experiment as well, then we would conclude that art style classification problems differ significantly from a typical object recognition problem and that there is room for improvement in this area.

References

Abadi, Martín, Agarwal, Ashish, Barham, Paul, Brevdo, Eugene, Chen, Zhifeng, Citro, Craig, Corrado, Greg S., Davis, Andy, Dean, Jeffrey, Devin, Matthieu, Ghemawat, Sanjay, Goodfellow, Ian, Harp, Andrew, Irving, Geoffrey, Isard, Michael, Jia, Yangqing, Jozefowicz, Rafal, Kaiser, Lukasz, Kudlur, Manjunath, Levenberg, Josh, Mané, Dan, Monga, Rajat, Moore, Sherry, Murray, Derek, Olah, Chris, Schuster, Mike, Shlens, Jonathon, Steiner, Benoit, Sutskever, Ilya, Talwar, Kunal, Tucker, Paul, Vanhoucke, Vincent, Vasudevan, Vijay, Viégas, Fernanda, Vinyals, Oriol, Warden, Pete, Wattenberg, Martin, Wicke, Martin, Yu, Yuan, and Zheng, Xiaoqiang. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

Bar, Yaniv, Levy, Noga, and Wolf, Lior. Classification of artistic styles using binarized features derived from a deep neural network.

Bickel, Steffen. Ecml-pkdd discovery challenge 2006 overview.

Bousmalis, Konstantinos, Trigeorgis, George, Silberman, Nathan, Krishnan, Dilip, and Erhan, Dumitru. Domain separation networks. In *Advances in Neural Information Processing Systems*, pp. 343–351, 2016.

Chollet, François et al. Keras. https://github.com/fchollet/keras, 2015.

Colyer, Adrian. Dynamic routing between capsules.

https://blog.acolyer.org/2017/11/13/dynamic-routing-between-capsules/. Accessed: Dec. 2017.

cs chan. Wikiart dataset. https://github.com/cs-chan/ICIP2016-PC/tree/master/WikiArt%20Dataset. ICIP2016-PC repository, Accessed: Dec. 2017.

Do, Chuong B and Ng, Andrew Y. Transfer learning for text classification. In *Advances in Neural Information Processing Systems*, pp. 299–306, 2006.

Domingos, Pedro. A few useful things to know about machine learning. *Communications of the ACM*, 55(10): 78–87, 2012.

Fernando, Chrisantha, Banarse, Dylan, Blundell, Charles, Zwols, Yori, Ha, David, Rusu, Andrei A, Pritzel, Alexander, and Wierstra, Daan. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

Fung, Vincent. An overview of resnet and its variants. https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035. Accessed: Dec. 2017.

Ganin, Yaroslav and Lempitsky, Victor. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pp. 1180–1189, 2015.

Guo, Xifeng. Xifengguo/capsnet-keras github repository. https://github.com/XifengGuo/CapsNet-Keras. Accessed: Dec. 2017.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385v1*, 2015.

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Johnson, C Richard, Hendriks, Ella, Berezhnoy, Igor J, Brevdo, Eugene, Hughes, Shannon M, Daubechies, Ingrid, Li, Jia, Postma, Eric, and Wang, James Z. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4), 2008.

Jou, Jonathan and Agrawal, Sandeep. Artist identification for renaissance paintings. 2012.

Kaiming He, Xiangyu Zhang, Shaoqing Ren Jian Sun. Identity mappings in deep residual networks.

Kotikalapudi, Raghavendra. Keras-vis documentation: Activation maximization. https://raghakot.github.io/keras-vis/visualizations/activation_maximization/. Accessed: Dec. 2017.

Kotikalapudi, Raghavendra and contributors. keras-vis. https://github.com/raghakot/keras-vis, 2017.

Lecoutre, Adrian, Negrevergne, Benjamin, and Yger, Florian. Recognizing art style automatically in painting with deep learning. In *Asian Conference on Machine Learning*, pp. 327–342, 2017.

Lombardi, Thomas Edward. *Classification of Style in Fine-art Painting*.

Mensink, Thomas and Van Gemert, Jan. The rijksmuseum challenge: Museum-centered visual recognition. In *Proceedings of International Conference on Multimedia Retrieval*, pp. 451. ACM, 2014.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Pechyonkin, Max. Understanding hintons capsule networks. part ii: How capsules work. https://medium.com/ai-theory-practice-business/understanding-hintons-capsule-networks-part-ii-ho Accessed: Dec. 2017.

Raina, Rajat, Ng, Andrew Y, and Koller, Daphne. Constructing informative priors using transfer learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 713–720. ACM, 2006.

Ruder, Sebastian and Plank, Barbara. Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*, 2017.

Ruder, Sebastian, Ghaffari, Parsa, and Breslin, John G. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv:1702.02052*, 2017.

Rusu, Andrei A, Rabinowitz, Neil C, Desjardins, Guillaume, Soyer, Hubert, Kirkpatrick, James, Kavukcuoglu, Koray, Pascanu, Razvan, and Hadsell, Raia. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Sablatnig, Robert, Kammerer, Paul, and Zolda, Ernestine. Hierarchical classification of paintings using face-and brush stroke models. In *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, pp. 172–174. IEEE, 1998.

Sabour, Sara, Frosst, Nicholas, and Hinton, Geoffrey E. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pp. 3859–3869, 2017.

Sener, Ozan, Song, Hyun Oh, Saxena, Ashutosh, and Savarese, Silvio. Learning transferrable representations for unsupervised domain adaptation. In *Advances in*

Neural Information Processing Systems, pp. 2110–2118, 2016.

Sharif Razavian, Ali, Azizpour, Hossein, Sullivan, Josephine, and Carlsson, Stefan. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

Tan, Wei Ren, Chan, Chee Seng, Aguirre, Hernán E., and Tanaka, Kiyoshi. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3703–3707, 2016.

Torrey, Lisa and Shavlik, Jude. Transfer learning. 2009.

Van Noord, Nanne, Hendriks, Ella, and Postma, Eric. Toward discovery of the artist's style: Learning to recognize artists by their artworks. *IEEE Signal Processing Magazine*, 32(4):46–54, 2015.

WikiArt. Wikipaintings. https://www.wikiart.org,. Accessed: Dec. 2017.

8. Appendix A

```
____________________________________________________________________________________________________

____________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
====================================================================================================
input_1 (InputLayer)            (None, 224, 224, 3)  0
____________________________________________________________________________________________________
conv1 (Conv2D)                  (None, 112, 112, 64) 9472        input_1[0][0]
____________________________________________________________________________________________________
bn_conv1 (BatchNormalization)   (None, 112, 112, 64) 256         conv1[0][0]
____________________________________________________________________________________________________
activation_1 (Activation)       (None, 112, 112, 64) 0           bn_conv1[0][0]
____________________________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 55, 55, 64)   0           activation_1[0][0]
____________________________________________________________________________________________________
res2a_branch2a (Conv2D)         (None, 55, 55, 64)   4160        max_pooling2d_1[0][0]
____________________________________________________________________________________________________
bn2a_branch2a (BatchNormalizati (None, 55, 55, 64)   256         res2a_branch2a[0][0]
____________________________________________________________________________________________________
activation_2 (Activation)       (None, 55, 55, 64)   0           bn2a_branch2a[0][0]
____________________________________________________________________________________________________
res2a_branch2b (Conv2D)         (None, 55, 55, 64)   36928       activation_2[0][0]
____________________________________________________________________________________________________
bn2a_branch2b (BatchNormalizati (None, 55, 55, 64)   256         res2a_branch2b[0][0]
____________________________________________________________________________________________________
activation_3 (Activation)       (None, 55, 55, 64)   0           bn2a_branch2b[0][0]
____________________________________________________________________________________________________
res2a_branch2c (Conv2D)         (None, 55, 55, 256)  16640       activation_3[0][0]
____________________________________________________________________________________________________
res2a_branch1 (Conv2D)          (None, 55, 55, 256)  16640       max_pooling2d_1[0][0]
____________________________________________________________________________________________________
bn2a_branch2c (BatchNormalizati (None, 55, 55, 256)  1024        res2a_branch2c[0][0]
____________________________________________________________________________________________________
bn2a_branch1 (BatchNormalizatio (None, 55, 55, 256)  1024        res2a_branch1[0][0]
____________________________________________________________________________________________________
add_1 (Add)                     (None, 55, 55, 256)  0           bn2a_branch2c[0][0]
                                                                 bn2a_branch1[0][0]
____________________________________________________________________________________________________
activation_4 (Activation)       (None, 55, 55, 256)  0           add_1[0][0]
____________________________________________________________________________________________________
res2b_branch2a (Conv2D)         (None, 55, 55, 64)   16448       activation_4[0][0]
____________________________________________________________________________________________________
bn2b_branch2a (BatchNormalizati (None, 55, 55, 64)   256         res2b_branch2a[0][0]
```

```
--------------------------------------------------------------------------------------------------
activation_5 (Activation)       (None, 55, 55, 64)   0            bn2b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res2b_branch2b (Conv2D)         (None, 55, 55, 64)   36928        activation_5[0][0]
--------------------------------------------------------------------------------------------------
bn2b_branch2b (BatchNormalizati (None, 55, 55, 64)   256          res2b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_6 (Activation)       (None, 55, 55, 64)   0            bn2b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res2b_branch2c (Conv2D)         (None, 55, 55, 256)  16640        activation_6[0][0]
--------------------------------------------------------------------------------------------------
bn2b_branch2c (BatchNormalizati (None, 55, 55, 256)  1024         res2b_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_2 (Add)                     (None, 55, 55, 256)  0            bn2b_branch2c[0][0]
                                                                  activation_4[0][0]
--------------------------------------------------------------------------------------------------
activation_7 (Activation)       (None, 55, 55, 256)  0            add_2[0][0]
--------------------------------------------------------------------------------------------------
res2c_branch2a (Conv2D)         (None, 55, 55, 64)   16448        activation_7[0][0]
--------------------------------------------------------------------------------------------------
bn2c_branch2a (BatchNormalizati (None, 55, 55, 64)   256          res2c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_8 (Activation)       (None, 55, 55, 64)   0            bn2c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res2c_branch2b (Conv2D)         (None, 55, 55, 64)   36928        activation_8[0][0]
--------------------------------------------------------------------------------------------------
bn2c_branch2b (BatchNormalizati (None, 55, 55, 64)   256          res2c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_9 (Activation)       (None, 55, 55, 64)   0            bn2c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res2c_branch2c (Conv2D)         (None, 55, 55, 256)  16640        activation_9[0][0]
--------------------------------------------------------------------------------------------------
bn2c_branch2c (BatchNormalizati (None, 55, 55, 256)  1024         res2c_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_3 (Add)                     (None, 55, 55, 256)  0            bn2c_branch2c[0][0]
                                                                  activation_7[0][0]
--------------------------------------------------------------------------------------------------
activation_10 (Activation)      (None, 55, 55, 256)  0            add_3[0][0]
--------------------------------------------------------------------------------------------------
res3a_branch2a (Conv2D)         (None, 28, 28, 128)  32896        activation_10[0][0]
--------------------------------------------------------------------------------------------------
bn3a_branch2a (BatchNormalizati (None, 28, 28, 128)  512          res3a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_11 (Activation)      (None, 28, 28, 128)  0            bn3a_branch2a[0][0]
```

```
________________________________________________________________________________________________
res3a_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_11[0][0]
________________________________________________________________________________________________
bn3a_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3a_branch2b[0][0]
________________________________________________________________________________________________
activation_12 (Activation)      (None, 28, 28, 128)  0           bn3a_branch2b[0][0]
________________________________________________________________________________________________
res3a_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_12[0][0]
________________________________________________________________________________________________
res3a_branch1 (Conv2D)          (None, 28, 28, 512)  131584      activation_10[0][0]
________________________________________________________________________________________________
bn3a_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3a_branch2c[0][0]
________________________________________________________________________________________________
bn3a_branch1 (BatchNormalizatio (None, 28, 28, 512)  2048        res3a_branch1[0][0]
________________________________________________________________________________________________
add_4 (Add)                     (None, 28, 28, 512)  0           bn3a_branch2c[0][0]
                                                                 bn3a_branch1[0][0]
________________________________________________________________________________________________
activation_13 (Activation)      (None, 28, 28, 512)  0           add_4[0][0]
________________________________________________________________________________________________
res3b_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_13[0][0]
________________________________________________________________________________________________
bn3b_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2a[0][0]
________________________________________________________________________________________________
activation_14 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2a[0][0]
________________________________________________________________________________________________
res3b_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_14[0][0]
________________________________________________________________________________________________
bn3b_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2b[0][0]
________________________________________________________________________________________________
activation_15 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2b[0][0]
________________________________________________________________________________________________
res3b_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_15[0][0]
________________________________________________________________________________________________
bn3b_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3b_branch2c[0][0]
________________________________________________________________________________________________
add_5 (Add)                     (None, 28, 28, 512)  0           bn3b_branch2c[0][0]
                                                                 activation_13[0][0]
________________________________________________________________________________________________
activation_16 (Activation)      (None, 28, 28, 512)  0           add_5[0][0]
________________________________________________________________________________________________
res3c_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_16[0][0]
________________________________________________________________________________________________
bn3c_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2a[0][0]
```

```
________________________________________________________________________________________________
activation_17 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2a[0][0]
________________________________________________________________________________________________
res3c_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_17[0][0]
________________________________________________________________________________________________
bn3c_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2b[0][0]
________________________________________________________________________________________________
activation_18 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2b[0][0]
________________________________________________________________________________________________
res3c_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_18[0][0]
________________________________________________________________________________________________
bn3c_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3c_branch2c[0][0]
________________________________________________________________________________________________
add_6 (Add)                     (None, 28, 28, 512)  0           bn3c_branch2c[0][0]
                                                                 activation_16[0][0]
________________________________________________________________________________________________
activation_19 (Activation)      (None, 28, 28, 512)  0           add_6[0][0]
________________________________________________________________________________________________
res3d_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_19[0][0]
________________________________________________________________________________________________
bn3d_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2a[0][0]
________________________________________________________________________________________________
activation_20 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2a[0][0]
________________________________________________________________________________________________
res3d_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_20[0][0]
________________________________________________________________________________________________
bn3d_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2b[0][0]
________________________________________________________________________________________________
activation_21 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2b[0][0]
________________________________________________________________________________________________
res3d_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_21[0][0]
________________________________________________________________________________________________
bn3d_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3d_branch2c[0][0]
________________________________________________________________________________________________
add_7 (Add)                     (None, 28, 28, 512)  0           bn3d_branch2c[0][0]
                                                                 activation_19[0][0]
________________________________________________________________________________________________
activation_22 (Activation)      (None, 28, 28, 512)  0           add_7[0][0]
________________________________________________________________________________________________
res4a_branch2a (Conv2D)         (None, 14, 14, 256)  131328      activation_22[0][0]
________________________________________________________________________________________________
bn4a_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4a_branch2a[0][0]
________________________________________________________________________________________________
activation_23 (Activation)      (None, 14, 14, 256)  0           bn4a_branch2a[0][0]
```

```
--------------------------------------------------------------------------------------------------
res4a_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_23[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_24 (Activation)       (None, 14, 14, 256)  0           bn4a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_24[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch1 (Conv2D)           (None, 14, 14, 1024) 525312      activation_22[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4a_branch2c[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch1 (BatchNormalizatio  (None, 14, 14, 1024) 4096        res4a_branch1[0][0]
--------------------------------------------------------------------------------------------------
add_8 (Add)                      (None, 14, 14, 1024) 0           bn4a_branch2c[0][0]
                                                                  bn4a_branch1[0][0]
--------------------------------------------------------------------------------------------------
activation_25 (Activation)       (None, 14, 14, 1024) 0           add_8[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2a (Conv2D)          (None, 14, 14, 256)  262400      activation_25[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_26 (Activation)       (None, 14, 14, 256)  0           bn4b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_26[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_27 (Activation)       (None, 14, 14, 256)  0           bn4b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_27[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4b_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_9 (Add)                      (None, 14, 14, 1024) 0           bn4b_branch2c[0][0]
                                                                  activation_25[0][0]
--------------------------------------------------------------------------------------------------
activation_28 (Activation)       (None, 14, 14, 1024) 0           add_9[0][0]
--------------------------------------------------------------------------------------------------
res4c_branch2a (Conv2D)          (None, 14, 14, 256)  262400      activation_28[0][0]
--------------------------------------------------------------------------------------------------
bn4c_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4c_branch2a[0][0]
```

```
__________________________________________________________________________________________________
activation_29 (Activation)      (None, 14, 14, 256)  0           bn4c_branch2a[0][0]
__________________________________________________________________________________________________
res4c_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_29[0][0]
__________________________________________________________________________________________________
bn4c_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4c_branch2b[0][0]
__________________________________________________________________________________________________
activation_30 (Activation)      (None, 14, 14, 256)  0           bn4c_branch2b[0][0]
__________________________________________________________________________________________________
res4c_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_30[0][0]
__________________________________________________________________________________________________
bn4c_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4c_branch2c[0][0]
__________________________________________________________________________________________________
add_10 (Add)                    (None, 14, 14, 1024) 0           bn4c_branch2c[0][0]
                                                                 activation_28[0][0]
__________________________________________________________________________________________________
activation_31 (Activation)      (None, 14, 14, 1024) 0           add_10[0][0]
__________________________________________________________________________________________________
res4d_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_31[0][0]
__________________________________________________________________________________________________
bn4d_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2a[0][0]
__________________________________________________________________________________________________
activation_32 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2a[0][0]
__________________________________________________________________________________________________
res4d_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_32[0][0]
__________________________________________________________________________________________________
bn4d_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2b[0][0]
__________________________________________________________________________________________________
activation_33 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2b[0][0]
__________________________________________________________________________________________________
res4d_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_33[0][0]
__________________________________________________________________________________________________
bn4d_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4d_branch2c[0][0]
__________________________________________________________________________________________________
add_11 (Add)                    (None, 14, 14, 1024) 0           bn4d_branch2c[0][0]
                                                                 activation_31[0][0]
__________________________________________________________________________________________________
activation_34 (Activation)      (None, 14, 14, 1024) 0           add_11[0][0]
__________________________________________________________________________________________________
res4e_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_34[0][0]
__________________________________________________________________________________________________
bn4e_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4e_branch2a[0][0]
__________________________________________________________________________________________________
activation_35 (Activation)      (None, 14, 14, 256)  0           bn4e_branch2a[0][0]
```

```
____________________________________________________________________________________________________
res4e_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_35[0][0]
____________________________________________________________________________________________________
bn4e_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4e_branch2b[0][0]
____________________________________________________________________________________________________
activation_36 (Activation)       (None, 14, 14, 256)  0           bn4e_branch2b[0][0]
____________________________________________________________________________________________________
res4e_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_36[0][0]
____________________________________________________________________________________________________
bn4e_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4e_branch2c[0][0]
____________________________________________________________________________________________________
add_12 (Add)                     (None, 14, 14, 1024) 0           bn4e_branch2c[0][0]
                                                                  activation_34[0][0]
____________________________________________________________________________________________________
activation_37 (Activation)       (None, 14, 14, 1024) 0           add_12[0][0]
____________________________________________________________________________________________________
res4f_branch2a (Conv2D)          (None, 14, 14, 256)  262400      activation_37[0][0]
____________________________________________________________________________________________________
bn4f_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4f_branch2a[0][0]
____________________________________________________________________________________________________
activation_38 (Activation)       (None, 14, 14, 256)  0           bn4f_branch2a[0][0]
____________________________________________________________________________________________________
res4f_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_38[0][0]
____________________________________________________________________________________________________
bn4f_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4f_branch2b[0][0]
____________________________________________________________________________________________________
activation_39 (Activation)       (None, 14, 14, 256)  0           bn4f_branch2b[0][0]
____________________________________________________________________________________________________
res4f_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_39[0][0]
____________________________________________________________________________________________________
bn4f_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4f_branch2c[0][0]
____________________________________________________________________________________________________
add_13 (Add)                     (None, 14, 14, 1024) 0           bn4f_branch2c[0][0]
                                                                  activation_37[0][0]
____________________________________________________________________________________________________
activation_40 (Activation)       (None, 14, 14, 1024) 0           add_13[0][0]
____________________________________________________________________________________________________
res5a_branch2a (Conv2D)          (None, 7, 7, 512)    524800      activation_40[0][0]
____________________________________________________________________________________________________
bn5a_branch2a (BatchNormalizati  (None, 7, 7, 512)    2048        res5a_branch2a[0][0]
____________________________________________________________________________________________________
activation_41 (Activation)       (None, 7, 7, 512)    0           bn5a_branch2a[0][0]
____________________________________________________________________________________________________
res5a_branch2b (Conv2D)          (None, 7, 7, 512)    2359808     activation_41[0][0]
```

```
----------------------------------------------------------------------------------------------------
bn5a_branch2b (BatchNormalizati (None, 7, 7, 512)    2048        res5a_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_42 (Activation)      (None, 7, 7, 512)    0           bn5a_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res5a_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624     activation_42[0][0]
----------------------------------------------------------------------------------------------------
res5a_branch1 (Conv2D)          (None, 7, 7, 2048)   2099200     activation_40[0][0]
----------------------------------------------------------------------------------------------------
bn5a_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5a_branch2c[0][0]
----------------------------------------------------------------------------------------------------
bn5a_branch1 (BatchNormalizatio (None, 7, 7, 2048)   8192        res5a_branch1[0][0]
----------------------------------------------------------------------------------------------------
add_14 (Add)                    (None, 7, 7, 2048)   0           bn5a_branch2c[0][0]
                                                                 bn5a_branch1[0][0]
----------------------------------------------------------------------------------------------------
activation_43 (Activation)      (None, 7, 7, 2048)   0           add_14[0][0]
----------------------------------------------------------------------------------------------------
res5b_branch2a (Conv2D)         (None, 7, 7, 512)    1049088     activation_43[0][0]
----------------------------------------------------------------------------------------------------
bn5b_branch2a (BatchNormalizati (None, 7, 7, 512)    2048        res5b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_44 (Activation)      (None, 7, 7, 512)    0           bn5b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
res5b_branch2b (Conv2D)         (None, 7, 7, 512)    2359808     activation_44[0][0]
----------------------------------------------------------------------------------------------------
bn5b_branch2b (BatchNormalizati (None, 7, 7, 512)    2048        res5b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_45 (Activation)      (None, 7, 7, 512)    0           bn5b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res5b_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624     activation_45[0][0]
----------------------------------------------------------------------------------------------------
bn5b_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5b_branch2c[0][0]
----------------------------------------------------------------------------------------------------
add_15 (Add)                    (None, 7, 7, 2048)   0           bn5b_branch2c[0][0]
                                                                 activation_43[0][0]
----------------------------------------------------------------------------------------------------
activation_46 (Activation)      (None, 7, 7, 2048)   0           add_15[0][0]
----------------------------------------------------------------------------------------------------
res5c_branch2a (Conv2D)         (None, 7, 7, 512)    1049088     activation_46[0][0]
----------------------------------------------------------------------------------------------------
bn5c_branch2a (BatchNormalizati (None, 7, 7, 512)    2048        res5c_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_47 (Activation)      (None, 7, 7, 512)    0           bn5c_branch2a[0][0]
```

```
________________________________________________________________________________
res5c_branch2b (Conv2D)         (None, 7, 7, 512)    2359808    activation_47[0][0]
________________________________________________________________________________
bn5c_branch2b (BatchNormalizati (None, 7, 7, 512)    2048       res5c_branch2b[0][0]
________________________________________________________________________________
activation_48 (Activation)      (None, 7, 7, 512)    0          bn5c_branch2b[0][0]
________________________________________________________________________________
res5c_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624    activation_48[0][0]
________________________________________________________________________________
bn5c_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192       res5c_branch2c[0][0]
________________________________________________________________________________
add_16 (Add)                    (None, 7, 7, 2048)   0          bn5c_branch2c[0][0]
                                                                activation_46[0][0]
________________________________________________________________________________
activation_49 (Activation)      (None, 7, 7, 2048)   0          add_16[0][0]
________________________________________________________________________________
avg_pool (AveragePooling2D)     (None, 1, 1, 2048)   0          activation_49[0][0]
________________________________________________________________________________
flatten_1 (Flatten)             (None, 2048)         0          avg_pool[0][0]
________________________________________________________________________________
output_predictions (Dense)      (None, 7)            14343      flatten_1[0][0]
================================================================================
Total params: 23,602,055
Trainable params: 23,548,935
Non-trainable params: 53,120
________________________________________________________________________________

________________________________________________________________________________
Layer (type)                    Output Shape         Param #    Connected to
================================================================================
input_1 (InputLayer)            (None, 224, 224, 3)  0
________________________________________________________________________________
conv1 (Conv2D)                  (None, 112, 112, 64) 9472       input_1[0][0]
________________________________________________________________________________
bn_conv1 (BatchNormalization)   (None, 112, 112, 64) 256        conv1[0][0]
________________________________________________________________________________
activation_1 (Activation)       (None, 112, 112, 64) 0          bn_conv1[0][0]
________________________________________________________________________________
max_pooling2d_1 (MaxPooling2D)  (None, 55, 55, 64)   0          activation_1[0][0]
________________________________________________________________________________
res2a_branch2a (Conv2D)         (None, 55, 55, 64)   4160       max_pooling2d_1[0][0]
________________________________________________________________________________
bn2a_branch2a (BatchNormalizati (None, 55, 55, 64)   256        res2a_branch2a[0][0]
________________________________________________________________________________
activation_2 (Activation)       (None, 55, 55, 64)   0          bn2a_branch2a[0][0]
```

```
__________________________________________________________________________________________________
res2a_branch2b (Conv2D)         (None, 55, 55, 64)   36928       activation_2[0][0]
__________________________________________________________________________________________________
bn2a_branch2b (BatchNormalizati (None, 55, 55, 64)   256         res2a_branch2b[0][0]
__________________________________________________________________________________________________
activation_3 (Activation)       (None, 55, 55, 64)   0           bn2a_branch2b[0][0]
__________________________________________________________________________________________________
res2a_branch2c (Conv2D)         (None, 55, 55, 256)  16640       activation_3[0][0]
__________________________________________________________________________________________________
res2a_branch1 (Conv2D)          (None, 55, 55, 256)  16640       max_pooling2d_1[0][0]
__________________________________________________________________________________________________
bn2a_branch2c (BatchNormalizati (None, 55, 55, 256)  1024        res2a_branch2c[0][0]
__________________________________________________________________________________________________
bn2a_branch1 (BatchNormalizatio (None, 55, 55, 256)  1024        res2a_branch1[0][0]
__________________________________________________________________________________________________
add_1 (Add)                     (None, 55, 55, 256)  0           bn2a_branch2c[0][0]
                                                                 bn2a_branch1[0][0]
__________________________________________________________________________________________________
activation_4 (Activation)       (None, 55, 55, 256)  0           add_1[0][0]
__________________________________________________________________________________________________
res2b_branch2a (Conv2D)         (None, 55, 55, 64)   16448       activation_4[0][0]
__________________________________________________________________________________________________
bn2b_branch2a (BatchNormalizati (None, 55, 55, 64)   256         res2b_branch2a[0][0]
__________________________________________________________________________________________________
activation_5 (Activation)       (None, 55, 55, 64)   0           bn2b_branch2a[0][0]
__________________________________________________________________________________________________
res2b_branch2b (Conv2D)         (None, 55, 55, 64)   36928       activation_5[0][0]
__________________________________________________________________________________________________
bn2b_branch2b (BatchNormalizati (None, 55, 55, 64)   256         res2b_branch2b[0][0]
__________________________________________________________________________________________________
activation_6 (Activation)       (None, 55, 55, 64)   0           bn2b_branch2b[0][0]
__________________________________________________________________________________________________
res2b_branch2c (Conv2D)         (None, 55, 55, 256)  16640       activation_6[0][0]
__________________________________________________________________________________________________
bn2b_branch2c (BatchNormalizati (None, 55, 55, 256)  1024        res2b_branch2c[0][0]
__________________________________________________________________________________________________
add_2 (Add)                     (None, 55, 55, 256)  0           bn2b_branch2c[0][0]
                                                                 activation_4[0][0]
__________________________________________________________________________________________________
activation_7 (Activation)       (None, 55, 55, 256)  0           add_2[0][0]
__________________________________________________________________________________________________
res2c_branch2a (Conv2D)         (None, 55, 55, 64)   16448       activation_7[0][0]
__________________________________________________________________________________________________
bn2c_branch2a (BatchNormalizati (None, 55, 55, 64)   256         res2c_branch2a[0][0]
```

```
____________________________________________________________________________________________________
activation_8 (Activation)       (None, 55, 55, 64)   0           bn2c_branch2a[0][0]
____________________________________________________________________________________________________
res2c_branch2b (Conv2D)         (None, 55, 55, 64)   36928       activation_8[0][0]
____________________________________________________________________________________________________
bn2c_branch2b (BatchNormalizati (None, 55, 55, 64)   256         res2c_branch2b[0][0]
____________________________________________________________________________________________________
activation_9 (Activation)       (None, 55, 55, 64)   0           bn2c_branch2b[0][0]
____________________________________________________________________________________________________
res2c_branch2c (Conv2D)         (None, 55, 55, 256)  16640       activation_9[0][0]
____________________________________________________________________________________________________
bn2c_branch2c (BatchNormalizati (None, 55, 55, 256)  1024        res2c_branch2c[0][0]
____________________________________________________________________________________________________
add_3 (Add)                     (None, 55, 55, 256)  0           bn2c_branch2c[0][0]
                                                                 activation_7[0][0]
____________________________________________________________________________________________________
activation_10 (Activation)      (None, 55, 55, 256)  0           add_3[0][0]
____________________________________________________________________________________________________
res3a_branch2a (Conv2D)         (None, 28, 28, 128)  32896       activation_10[0][0]
____________________________________________________________________________________________________
bn3a_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3a_branch2a[0][0]
____________________________________________________________________________________________________
activation_11 (Activation)      (None, 28, 28, 128)  0           bn3a_branch2a[0][0]
____________________________________________________________________________________________________
res3a_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_11[0][0]
____________________________________________________________________________________________________
bn3a_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3a_branch2b[0][0]
____________________________________________________________________________________________________
activation_12 (Activation)      (None, 28, 28, 128)  0           bn3a_branch2b[0][0]
____________________________________________________________________________________________________
res3a_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_12[0][0]
____________________________________________________________________________________________________
res3a_branch1 (Conv2D)          (None, 28, 28, 512)  131584      activation_10[0][0]
____________________________________________________________________________________________________
bn3a_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3a_branch2c[0][0]
____________________________________________________________________________________________________
bn3a_branch1 (BatchNormalizatio (None, 28, 28, 512)  2048        res3a_branch1[0][0]
____________________________________________________________________________________________________
add_4 (Add)                     (None, 28, 28, 512)  0           bn3a_branch2c[0][0]
                                                                 bn3a_branch1[0][0]
____________________________________________________________________________________________________
activation_13 (Activation)      (None, 28, 28, 512)  0           add_4[0][0]
____________________________________________________________________________________________________
res3b_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_13[0][0]
```

```
----------------------------------------------------------------------------------------------------
bn3b_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_14 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
res3b_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_14[0][0]
----------------------------------------------------------------------------------------------------
bn3b_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_15 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res3b_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_15[0][0]
----------------------------------------------------------------------------------------------------
bn3b_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3b_branch2c[0][0]
----------------------------------------------------------------------------------------------------
add_5 (Add)                     (None, 28, 28, 512)  0           bn3b_branch2c[0][0]
                                                                 activation_13[0][0]
----------------------------------------------------------------------------------------------------
activation_16 (Activation)      (None, 28, 28, 512)  0           add_5[0][0]
----------------------------------------------------------------------------------------------------
res3c_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_16[0][0]
----------------------------------------------------------------------------------------------------
bn3c_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_17 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2a[0][0]
----------------------------------------------------------------------------------------------------
res3c_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_17[0][0]
----------------------------------------------------------------------------------------------------
bn3c_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_18 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res3c_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_18[0][0]
----------------------------------------------------------------------------------------------------
bn3c_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3c_branch2c[0][0]
----------------------------------------------------------------------------------------------------
add_6 (Add)                     (None, 28, 28, 512)  0           bn3c_branch2c[0][0]
                                                                 activation_16[0][0]
----------------------------------------------------------------------------------------------------
activation_19 (Activation)      (None, 28, 28, 512)  0           add_6[0][0]
----------------------------------------------------------------------------------------------------
res3d_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_19[0][0]
----------------------------------------------------------------------------------------------------
bn3d_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2a[0][0]
```

```
__________________________________________________________________________________________________
activation_20 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2a[0][0]
__________________________________________________________________________________________________
res3d_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_20[0][0]
__________________________________________________________________________________________________
bn3d_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2b[0][0]
__________________________________________________________________________________________________
activation_21 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2b[0][0]
__________________________________________________________________________________________________
res3d_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_21[0][0]
__________________________________________________________________________________________________
bn3d_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3d_branch2c[0][0]
__________________________________________________________________________________________________
add_7 (Add)                     (None, 28, 28, 512)  0           bn3d_branch2c[0][0]
                                                                 activation_19[0][0]
__________________________________________________________________________________________________
activation_22 (Activation)      (None, 28, 28, 512)  0           add_7[0][0]
__________________________________________________________________________________________________
res4a_branch2a (Conv2D)         (None, 14, 14, 256)  131328      activation_22[0][0]
__________________________________________________________________________________________________
bn4a_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4a_branch2a[0][0]
__________________________________________________________________________________________________
activation_23 (Activation)      (None, 14, 14, 256)  0           bn4a_branch2a[0][0]
__________________________________________________________________________________________________
res4a_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_23[0][0]
__________________________________________________________________________________________________
bn4a_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4a_branch2b[0][0]
__________________________________________________________________________________________________
activation_24 (Activation)      (None, 14, 14, 256)  0           bn4a_branch2b[0][0]
__________________________________________________________________________________________________
res4a_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_24[0][0]
__________________________________________________________________________________________________
res4a_branch1 (Conv2D)          (None, 14, 14, 1024) 525312      activation_22[0][0]
__________________________________________________________________________________________________
bn4a_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4a_branch2c[0][0]
__________________________________________________________________________________________________
bn4a_branch1 (BatchNormalizatio (None, 14, 14, 1024) 4096        res4a_branch1[0][0]
__________________________________________________________________________________________________
add_8 (Add)                     (None, 14, 14, 1024) 0           bn4a_branch2c[0][0]
                                                                 bn4a_branch1[0][0]
__________________________________________________________________________________________________
activation_25 (Activation)      (None, 14, 14, 1024) 0           add_8[0][0]
__________________________________________________________________________________________________
res4b_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_25[0][0]
```

```
----------------------------------------------------------------------------------------------------
bn4b_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_26 (Activation)      (None, 14, 14, 256)  0           bn4b_branch2a[0][0]
----------------------------------------------------------------------------------------------------
res4b_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_26[0][0]
----------------------------------------------------------------------------------------------------
bn4b_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_27 (Activation)      (None, 14, 14, 256)  0           bn4b_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res4b_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_27[0][0]
----------------------------------------------------------------------------------------------------
bn4b_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4b_branch2c[0][0]
----------------------------------------------------------------------------------------------------
add_9 (Add)                     (None, 14, 14, 1024) 0           bn4b_branch2c[0][0]
                                                                 activation_25[0][0]
----------------------------------------------------------------------------------------------------
activation_28 (Activation)      (None, 14, 14, 1024) 0           add_9[0][0]
----------------------------------------------------------------------------------------------------
res4c_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_28[0][0]
----------------------------------------------------------------------------------------------------
bn4c_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4c_branch2a[0][0]
----------------------------------------------------------------------------------------------------
activation_29 (Activation)      (None, 14, 14, 256)  0           bn4c_branch2a[0][0]
----------------------------------------------------------------------------------------------------
res4c_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_29[0][0]
----------------------------------------------------------------------------------------------------
bn4c_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4c_branch2b[0][0]
----------------------------------------------------------------------------------------------------
activation_30 (Activation)      (None, 14, 14, 256)  0           bn4c_branch2b[0][0]
----------------------------------------------------------------------------------------------------
res4c_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_30[0][0]
----------------------------------------------------------------------------------------------------
bn4c_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4c_branch2c[0][0]
----------------------------------------------------------------------------------------------------
add_10 (Add)                    (None, 14, 14, 1024) 0           bn4c_branch2c[0][0]
                                                                 activation_28[0][0]
----------------------------------------------------------------------------------------------------
activation_31 (Activation)      (None, 14, 14, 1024) 0           add_10[0][0]
----------------------------------------------------------------------------------------------------
res4d_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_31[0][0]
----------------------------------------------------------------------------------------------------
bn4d_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2a[0][0]
```

```
____________________________________________________________________________________________________
activation_32 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2a[0][0]
____________________________________________________________________________________________________
res4d_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_32[0][0]
____________________________________________________________________________________________________
bn4d_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2b[0][0]
____________________________________________________________________________________________________
activation_33 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2b[0][0]
____________________________________________________________________________________________________
res4d_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_33[0][0]
____________________________________________________________________________________________________
bn4d_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4d_branch2c[0][0]
____________________________________________________________________________________________________
add_11 (Add)                    (None, 14, 14, 1024) 0           bn4d_branch2c[0][0]
                                                                 activation_31[0][0]
____________________________________________________________________________________________________
activation_34 (Activation)      (None, 14, 14, 1024) 0           add_11[0][0]
____________________________________________________________________________________________________
res4e_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_34[0][0]
____________________________________________________________________________________________________
bn4e_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4e_branch2a[0][0]
____________________________________________________________________________________________________
activation_35 (Activation)      (None, 14, 14, 256)  0           bn4e_branch2a[0][0]
____________________________________________________________________________________________________
res4e_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_35[0][0]
____________________________________________________________________________________________________
bn4e_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4e_branch2b[0][0]
____________________________________________________________________________________________________
activation_36 (Activation)      (None, 14, 14, 256)  0           bn4e_branch2b[0][0]
____________________________________________________________________________________________________
res4e_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_36[0][0]
____________________________________________________________________________________________________
bn4e_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4e_branch2c[0][0]
____________________________________________________________________________________________________
add_12 (Add)                    (None, 14, 14, 1024) 0           bn4e_branch2c[0][0]
                                                                 activation_34[0][0]
____________________________________________________________________________________________________
activation_37 (Activation)      (None, 14, 14, 1024) 0           add_12[0][0]
____________________________________________________________________________________________________
res4f_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_37[0][0]
____________________________________________________________________________________________________
bn4f_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4f_branch2a[0][0]
____________________________________________________________________________________________________
activation_38 (Activation)      (None, 14, 14, 256)  0           bn4f_branch2a[0][0]
```

```
__________________________________________________________________________________________________
res4f_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_38[0][0]
__________________________________________________________________________________________________
bn4f_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4f_branch2b[0][0]
__________________________________________________________________________________________________
activation_39 (Activation)      (None, 14, 14, 256)  0           bn4f_branch2b[0][0]
__________________________________________________________________________________________________
res4f_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_39[0][0]
__________________________________________________________________________________________________
bn4f_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4f_branch2c[0][0]
__________________________________________________________________________________________________
add_13 (Add)                    (None, 14, 14, 1024) 0           bn4f_branch2c[0][0]
                                                                 activation_37[0][0]
__________________________________________________________________________________________________
activation_40 (Activation)      (None, 14, 14, 1024) 0           add_13[0][0]
__________________________________________________________________________________________________
res5a_branch2a (Conv2D)         (None, 7, 7, 512)    524800      activation_40[0][0]
__________________________________________________________________________________________________
bn5a_branch2a (BatchNormalizati (None, 7, 7, 512)    2048        res5a_branch2a[0][0]
__________________________________________________________________________________________________
activation_41 (Activation)      (None, 7, 7, 512)    0           bn5a_branch2a[0][0]
__________________________________________________________________________________________________
res5a_branch2b (Conv2D)         (None, 7, 7, 512)    2359808     activation_41[0][0]
__________________________________________________________________________________________________
bn5a_branch2b (BatchNormalizati (None, 7, 7, 512)    2048        res5a_branch2b[0][0]
__________________________________________________________________________________________________
activation_42 (Activation)      (None, 7, 7, 512)    0           bn5a_branch2b[0][0]
__________________________________________________________________________________________________
res5a_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624     activation_42[0][0]
__________________________________________________________________________________________________
res5a_branch1 (Conv2D)          (None, 7, 7, 2048)   2099200     activation_40[0][0]
__________________________________________________________________________________________________
bn5a_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5a_branch2c[0][0]
__________________________________________________________________________________________________
bn5a_branch1 (BatchNormalizatio (None, 7, 7, 2048)   8192        res5a_branch1[0][0]
__________________________________________________________________________________________________
add_14 (Add)                    (None, 7, 7, 2048)   0           bn5a_branch2c[0][0]
                                                                 bn5a_branch1[0][0]
__________________________________________________________________________________________________
activation_43 (Activation)      (None, 7, 7, 2048)   0           add_14[0][0]
__________________________________________________________________________________________________
res5b_branch2a (Conv2D)         (None, 7, 7, 512)    1049088     activation_43[0][0]
__________________________________________________________________________________________________
bn5b_branch2a (BatchNormalizati (None, 7, 7, 512)    2048        res5b_branch2a[0][0]
```

```
________________________________________________________________________________________________
activation_44 (Activation)      (None, 7, 7, 512)    0           bn5b_branch2a[0][0]
________________________________________________________________________________________________
res5b_branch2b (Conv2D)         (None, 7, 7, 512)    2359808     activation_44[0][0]
________________________________________________________________________________________________
bn5b_branch2b (BatchNormalizati (None, 7, 7, 512)    2048        res5b_branch2b[0][0]
________________________________________________________________________________________________
activation_45 (Activation)      (None, 7, 7, 512)    0           bn5b_branch2b[0][0]
________________________________________________________________________________________________
res5b_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624     activation_45[0][0]
________________________________________________________________________________________________
bn5b_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5b_branch2c[0][0]
________________________________________________________________________________________________
add_15 (Add)                    (None, 7, 7, 2048)   0           bn5b_branch2c[0][0]
                                                                 activation_43[0][0]
________________________________________________________________________________________________
activation_46 (Activation)      (None, 7, 7, 2048)   0           add_15[0][0]
________________________________________________________________________________________________
res5c_branch2a (Conv2D)         (None, 7, 7, 512)    1049088     activation_46[0][0]
________________________________________________________________________________________________
bn5c_branch2a (BatchNormalizati (None, 7, 7, 512)    2048        res5c_branch2a[0][0]
________________________________________________________________________________________________
activation_47 (Activation)      (None, 7, 7, 512)    0           bn5c_branch2a[0][0]
________________________________________________________________________________________________
res5c_branch2b (Conv2D)         (None, 7, 7, 512)    2359808     activation_47[0][0]
________________________________________________________________________________________________
bn5c_branch2b (BatchNormalizati (None, 7, 7, 512)    2048        res5c_branch2b[0][0]
________________________________________________________________________________________________
activation_48 (Activation)      (None, 7, 7, 512)    0           bn5c_branch2b[0][0]
________________________________________________________________________________________________
res5c_branch2c (Conv2D)         (None, 7, 7, 2048)   1050624     activation_48[0][0]
________________________________________________________________________________________________
bn5c_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5c_branch2c[0][0]
________________________________________________________________________________________________
add_16 (Add)                    (None, 7, 7, 2048)   0           bn5c_branch2c[0][0]
                                                                 activation_46[0][0]
________________________________________________________________________________________________
activation_49 (Activation)      (None, 7, 7, 2048)   0           add_16[0][0]
________________________________________________________________________________________________
avg_pool (AveragePooling2D)     (None, 1, 1, 2048)   0           activation_49[0][0]
________________________________________________________________________________________________
flatten_1 (Flatten)             (None, 2048)         0           avg_pool[0][0]
________________________________________________________________________________________________
output_predictions (Dense)      (None, 7)            14343       flatten_1[0][0]
```

```
==================================================================================================

Total params: 23,602,055
Trainable params: 23,548,935
Non-trainable params: 53,120
--------------------------------------------------------------------------------------------------
```

Listing 2. Resnet50 Model Summary

```
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 100, 100, 3)       0
-----------------------------------------------------------------
conv1 (Conv2D)               (None, 92, 92, 256)       62464
-----------------------------------------------------------------
conv2d_1 (Conv2D)            (None, 42, 42, 256)       5308672
-----------------------------------------------------------------
reshape_1 (Reshape)          (None, 56448, 8)          0
-----------------------------------------------------------------
lambda_1 (Lambda)            (None, 56448, 8)          0
-----------------------------------------------------------------
digitcaps (CapsuleLayer)     (None, 8, 16)             58254336
-----------------------------------------------------------------
out_caps (Length)            (None, 8)                 0
=================================================================
Total params: 63,625,472
Trainable params: 63,173,888
Non-trainable params: 451,584
-----------------------------------------------------------------
```

Listing 3. Capsnet Model Summary

```
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_1 (InputLayer)            (None, 224, 224, 3)  0
--------------------------------------------------------------------------------------------------
conv1 (Conv2D)                  (None, 112, 112, 64) 9472        input_1[0][0]
--------------------------------------------------------------------------------------------------
bn_conv1 (BatchNormalization)   (None, 112, 112, 64) 256         conv1[0][0]
--------------------------------------------------------------------------------------------------
activation_1 (Activation)       (None, 112, 112, 64) 0           bn_conv1[0][0]
--------------------------------------------------------------------------------------------------
max_pooling2d_1 (MaxPooling2D)  (None, 55, 55, 64)   0           activation_1[0][0]
--------------------------------------------------------------------------------------------------
res2a_branch2a (Conv2D)         (None, 55, 55, 64)   4160        max_pooling2d_1[0][0]
--------------------------------------------------------------------------------------------------
bn2a_branch2a (BatchNormalizati (None, 55, 55, 64)   256         res2a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_2 (Activation)       (None, 55, 55, 64)   0           bn2a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res2a_branch2b (Conv2D)         (None, 55, 55, 64)   36928       activation_2[0][0]
--------------------------------------------------------------------------------------------------
bn2a_branch2b (BatchNormalizati (None, 55, 55, 64)   256         res2a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_3 (Activation)       (None, 55, 55, 64)   0           bn2a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res2a_branch2c (Conv2D)         (None, 55, 55, 256)  16640       activation_3[0][0]
--------------------------------------------------------------------------------------------------
res2a_branch1 (Conv2D)          (None, 55, 55, 256)  16640       max_pooling2d_1[0][0]
--------------------------------------------------------------------------------------------------
bn2a_branch2c (BatchNormalizati (None, 55, 55, 256)  1024        res2a_branch2c[0][0]
--------------------------------------------------------------------------------------------------
bn2a_branch1 (BatchNormalizatio (None, 55, 55, 256)  1024        res2a_branch1[0][0]
```

```
____________________________________________________________________________________________________
add_1 (Add)                       (None, 55, 55, 256)  0            bn2a_branch2c[0][0]
                                                                    bn2a_branch1[0][0]
____________________________________________________________________________________________________
activation_4 (Activation)         (None, 55, 55, 256)  0            add_1[0][0]
____________________________________________________________________________________________________
res2b_branch2a (Conv2D)           (None, 55, 55, 64)   16448        activation_4[0][0]
____________________________________________________________________________________________________
bn2b_branch2a (BatchNormalizati   (None, 55, 55, 64)   256          res2b_branch2a[0][0]
____________________________________________________________________________________________________
activation_5 (Activation)         (None, 55, 55, 64)   0            bn2b_branch2a[0][0]
____________________________________________________________________________________________________
res2b_branch2b (Conv2D)           (None, 55, 55, 64)   36928        activation_5[0][0]
____________________________________________________________________________________________________
bn2b_branch2b (BatchNormalizati   (None, 55, 55, 64)   256          res2b_branch2b[0][0]
____________________________________________________________________________________________________
activation_6 (Activation)         (None, 55, 55, 64)   0            bn2b_branch2b[0][0]
____________________________________________________________________________________________________
res2b_branch2c (Conv2D)           (None, 55, 55, 256)  16640        activation_6[0][0]
____________________________________________________________________________________________________
bn2b_branch2c (BatchNormalizati   (None, 55, 55, 256)  1024         res2b_branch2c[0][0]
____________________________________________________________________________________________________
add_2 (Add)                       (None, 55, 55, 256)  0            bn2b_branch2c[0][0]
                                                                    activation_4[0][0]
____________________________________________________________________________________________________
activation_7 (Activation)         (None, 55, 55, 256)  0            add_2[0][0]
____________________________________________________________________________________________________
res2c_branch2a (Conv2D)           (None, 55, 55, 64)   16448        activation_7[0][0]
____________________________________________________________________________________________________
bn2c_branch2a (BatchNormalizati   (None, 55, 55, 64)   256          res2c_branch2a[0][0]
____________________________________________________________________________________________________
activation_8 (Activation)         (None, 55, 55, 64)   0            bn2c_branch2a[0][0]
____________________________________________________________________________________________________
res2c_branch2b (Conv2D)           (None, 55, 55, 64)   36928        activation_8[0][0]
____________________________________________________________________________________________________
bn2c_branch2b (BatchNormalizati   (None, 55, 55, 64)   256          res2c_branch2b[0][0]
____________________________________________________________________________________________________
activation_9 (Activation)         (None, 55, 55, 64)   0            bn2c_branch2b[0][0]
____________________________________________________________________________________________________
res2c_branch2c (Conv2D)           (None, 55, 55, 256)  16640        activation_9[0][0]
____________________________________________________________________________________________________
bn2c_branch2c (BatchNormalizati   (None, 55, 55, 256)  1024         res2c_branch2c[0][0]
____________________________________________________________________________________________________
add_3 (Add)                       (None, 55, 55, 256)  0            bn2c_branch2c[0][0]
                                                                    activation_7[0][0]
____________________________________________________________________________________________________
activation_10 (Activation)        (None, 55, 55, 256)  0            add_3[0][0]
____________________________________________________________________________________________________
res3a_branch2a (Conv2D)           (None, 28, 28, 128)  32896        activation_10[0][0]
____________________________________________________________________________________________________
bn3a_branch2a (BatchNormalizati   (None, 28, 28, 128)  512          res3a_branch2a[0][0]
____________________________________________________________________________________________________
activation_11 (Activation)        (None, 28, 28, 128)  0            bn3a_branch2a[0][0]
____________________________________________________________________________________________________
res3a_branch2b (Conv2D)           (None, 28, 28, 128)  147584       activation_11[0][0]
____________________________________________________________________________________________________
bn3a_branch2b (BatchNormalizati   (None, 28, 28, 128)  512          res3a_branch2b[0][0]
____________________________________________________________________________________________________
activation_12 (Activation)        (None, 28, 28, 128)  0            bn3a_branch2b[0][0]
____________________________________________________________________________________________________
res3a_branch2c (Conv2D)           (None, 28, 28, 512)  66048        activation_12[0][0]
____________________________________________________________________________________________________
res3a_branch1 (Conv2D)            (None, 28, 28, 512)  131584       activation_10[0][0]
____________________________________________________________________________________________________
bn3a_branch2c (BatchNormalizati   (None, 28, 28, 512)  2048         res3a_branch2c[0][0]
```

```
________________________________________________________________________________________________
bn3a_branch1 (BatchNormalizatio (None, 28, 28, 512)  2048        res3a_branch1[0][0]
________________________________________________________________________________________________
add_4 (Add)                     (None, 28, 28, 512)  0           bn3a_branch2c[0][0]
                                                                 bn3a_branch1[0][0]
________________________________________________________________________________________________
activation_13 (Activation)      (None, 28, 28, 512)  0           add_4[0][0]
________________________________________________________________________________________________
res3b_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_13[0][0]
________________________________________________________________________________________________
bn3b_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2a[0][0]
________________________________________________________________________________________________
activation_14 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2a[0][0]
________________________________________________________________________________________________
res3b_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_14[0][0]
________________________________________________________________________________________________
bn3b_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3b_branch2b[0][0]
________________________________________________________________________________________________
activation_15 (Activation)      (None, 28, 28, 128)  0           bn3b_branch2b[0][0]
________________________________________________________________________________________________
res3b_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_15[0][0]
________________________________________________________________________________________________
bn3b_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3b_branch2c[0][0]
________________________________________________________________________________________________
add_5 (Add)                     (None, 28, 28, 512)  0           bn3b_branch2c[0][0]
                                                                 activation_13[0][0]
________________________________________________________________________________________________
activation_16 (Activation)      (None, 28, 28, 512)  0           add_5[0][0]
________________________________________________________________________________________________
res3c_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_16[0][0]
________________________________________________________________________________________________
bn3c_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2a[0][0]
________________________________________________________________________________________________
activation_17 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2a[0][0]
________________________________________________________________________________________________
res3c_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_17[0][0]
________________________________________________________________________________________________
bn3c_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3c_branch2b[0][0]
________________________________________________________________________________________________
activation_18 (Activation)      (None, 28, 28, 128)  0           bn3c_branch2b[0][0]
________________________________________________________________________________________________
res3c_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_18[0][0]
________________________________________________________________________________________________
bn3c_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3c_branch2c[0][0]
________________________________________________________________________________________________
add_6 (Add)                     (None, 28, 28, 512)  0           bn3c_branch2c[0][0]
                                                                 activation_16[0][0]
________________________________________________________________________________________________
activation_19 (Activation)      (None, 28, 28, 512)  0           add_6[0][0]
________________________________________________________________________________________________
res3d_branch2a (Conv2D)         (None, 28, 28, 128)  65664       activation_19[0][0]
________________________________________________________________________________________________
bn3d_branch2a (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2a[0][0]
________________________________________________________________________________________________
activation_20 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2a[0][0]
________________________________________________________________________________________________
res3d_branch2b (Conv2D)         (None, 28, 28, 128)  147584      activation_20[0][0]
________________________________________________________________________________________________
bn3d_branch2b (BatchNormalizati (None, 28, 28, 128)  512         res3d_branch2b[0][0]
________________________________________________________________________________________________
activation_21 (Activation)      (None, 28, 28, 128)  0           bn3d_branch2b[0][0]
________________________________________________________________________________________________
res3d_branch2c (Conv2D)         (None, 28, 28, 512)  66048       activation_21[0][0]
________________________________________________________________________________________________
bn3d_branch2c (BatchNormalizati (None, 28, 28, 512)  2048        res3d_branch2c[0][0]
```

```
--------------------------------------------------------------------------------------------------
add_7 (Add)                      (None, 28, 28, 512)  0           bn3d_branch2c[0][0]
                                                                  activation_19[0][0]
--------------------------------------------------------------------------------------------------
activation_22 (Activation)       (None, 28, 28, 512)  0           add_7[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch2a (Conv2D)          (None, 14, 14, 256)  131328      activation_22[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_23 (Activation)       (None, 14, 14, 256)  0           bn4a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_23[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_24 (Activation)       (None, 14, 14, 256)  0           bn4a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_24[0][0]
--------------------------------------------------------------------------------------------------
res4a_branch1 (Conv2D)           (None, 14, 14, 1024) 525312      activation_22[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4a_branch2c[0][0]
--------------------------------------------------------------------------------------------------
bn4a_branch1 (BatchNormalizatio  (None, 14, 14, 1024) 4096        res4a_branch1[0][0]
--------------------------------------------------------------------------------------------------
add_8 (Add)                      (None, 14, 14, 1024) 0           bn4a_branch2c[0][0]
                                                                  bn4a_branch1[0][0]
--------------------------------------------------------------------------------------------------
activation_25 (Activation)       (None, 14, 14, 1024) 0           add_8[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2a (Conv2D)          (None, 14, 14, 256)  262400      activation_25[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_26 (Activation)       (None, 14, 14, 256)  0           bn4b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_26[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_27 (Activation)       (None, 14, 14, 256)  0           bn4b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res4b_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_27[0][0]
--------------------------------------------------------------------------------------------------
bn4b_branch2c (BatchNormalizati  (None, 14, 14, 1024) 4096        res4b_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_9 (Add)                      (None, 14, 14, 1024) 0           bn4b_branch2c[0][0]
                                                                  activation_25[0][0]
--------------------------------------------------------------------------------------------------
activation_28 (Activation)       (None, 14, 14, 1024) 0           add_9[0][0]
--------------------------------------------------------------------------------------------------
res4c_branch2a (Conv2D)          (None, 14, 14, 256)  262400      activation_28[0][0]
--------------------------------------------------------------------------------------------------
bn4c_branch2a (BatchNormalizati  (None, 14, 14, 256)  1024        res4c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_29 (Activation)       (None, 14, 14, 256)  0           bn4c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res4c_branch2b (Conv2D)          (None, 14, 14, 256)  590080      activation_29[0][0]
--------------------------------------------------------------------------------------------------
bn4c_branch2b (BatchNormalizati  (None, 14, 14, 256)  1024        res4c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_30 (Activation)       (None, 14, 14, 256)  0           bn4c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res4c_branch2c (Conv2D)          (None, 14, 14, 1024) 263168      activation_30[0][0]
```

```
________________________________________________________________________________________________
bn4c_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4c_branch2c[0][0]
________________________________________________________________________________________________
add_10 (Add)                    (None, 14, 14, 1024) 0           bn4c_branch2c[0][0]
                                                                 activation_28[0][0]
________________________________________________________________________________________________
activation_31 (Activation)      (None, 14, 14, 1024) 0           add_10[0][0]
________________________________________________________________________________________________
res4d_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_31[0][0]
________________________________________________________________________________________________
bn4d_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2a[0][0]
________________________________________________________________________________________________
activation_32 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2a[0][0]
________________________________________________________________________________________________
res4d_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_32[0][0]
________________________________________________________________________________________________
bn4d_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4d_branch2b[0][0]
________________________________________________________________________________________________
activation_33 (Activation)      (None, 14, 14, 256)  0           bn4d_branch2b[0][0]
________________________________________________________________________________________________
res4d_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_33[0][0]
________________________________________________________________________________________________
bn4d_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4d_branch2c[0][0]
________________________________________________________________________________________________
add_11 (Add)                    (None, 14, 14, 1024) 0           bn4d_branch2c[0][0]
                                                                 activation_31[0][0]
________________________________________________________________________________________________
activation_34 (Activation)      (None, 14, 14, 1024) 0           add_11[0][0]
________________________________________________________________________________________________
res4e_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_34[0][0]
________________________________________________________________________________________________
bn4e_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4e_branch2a[0][0]
________________________________________________________________________________________________
activation_35 (Activation)      (None, 14, 14, 256)  0           bn4e_branch2a[0][0]
________________________________________________________________________________________________
res4e_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_35[0][0]
________________________________________________________________________________________________
bn4e_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4e_branch2b[0][0]
________________________________________________________________________________________________
activation_36 (Activation)      (None, 14, 14, 256)  0           bn4e_branch2b[0][0]
________________________________________________________________________________________________
res4e_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_36[0][0]
________________________________________________________________________________________________
bn4e_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4e_branch2c[0][0]
________________________________________________________________________________________________
add_12 (Add)                    (None, 14, 14, 1024) 0           bn4e_branch2c[0][0]
                                                                 activation_34[0][0]
________________________________________________________________________________________________
activation_37 (Activation)      (None, 14, 14, 1024) 0           add_12[0][0]
________________________________________________________________________________________________
res4f_branch2a (Conv2D)         (None, 14, 14, 256)  262400      activation_37[0][0]
________________________________________________________________________________________________
bn4f_branch2a (BatchNormalizati (None, 14, 14, 256)  1024        res4f_branch2a[0][0]
________________________________________________________________________________________________
activation_38 (Activation)      (None, 14, 14, 256)  0           bn4f_branch2a[0][0]
________________________________________________________________________________________________
res4f_branch2b (Conv2D)         (None, 14, 14, 256)  590080      activation_38[0][0]
________________________________________________________________________________________________
bn4f_branch2b (BatchNormalizati (None, 14, 14, 256)  1024        res4f_branch2b[0][0]
________________________________________________________________________________________________
activation_39 (Activation)      (None, 14, 14, 256)  0           bn4f_branch2b[0][0]
________________________________________________________________________________________________
res4f_branch2c (Conv2D)         (None, 14, 14, 1024) 263168      activation_39[0][0]
________________________________________________________________________________________________
bn4f_branch2c (BatchNormalizati (None, 14, 14, 1024) 4096        res4f_branch2c[0][0]
```

```
--------------------------------------------------------------------------------------------------
add_13 (Add)                     (None, 14, 14, 1024) 0           bn4f_branch2c[0][0]
                                                                  activation_37[0][0]
--------------------------------------------------------------------------------------------------
activation_40 (Activation)       (None, 14, 14, 1024) 0           add_13[0][0]
--------------------------------------------------------------------------------------------------
res5a_branch2a (Conv2D)          (None, 7, 7, 512)    524800      activation_40[0][0]
--------------------------------------------------------------------------------------------------
bn5a_branch2a (BatchNormalizati  (None, 7, 7, 512)    2048        res5a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_41 (Activation)       (None, 7, 7, 512)    0           bn5a_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res5a_branch2b (Conv2D)          (None, 7, 7, 512)    2359808     activation_41[0][0]
--------------------------------------------------------------------------------------------------
bn5a_branch2b (BatchNormalizati  (None, 7, 7, 512)    2048        res5a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_42 (Activation)       (None, 7, 7, 512)    0           bn5a_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res5a_branch2c (Conv2D)          (None, 7, 7, 2048)   1050624     activation_42[0][0]
--------------------------------------------------------------------------------------------------
res5a_branch1 (Conv2D)           (None, 7, 7, 2048)   2099200     activation_40[0][0]
--------------------------------------------------------------------------------------------------
bn5a_branch2c (BatchNormalizati  (None, 7, 7, 2048)   8192        res5a_branch2c[0][0]
--------------------------------------------------------------------------------------------------
bn5a_branch1 (BatchNormalizatio  (None, 7, 7, 2048)   8192        res5a_branch1[0][0]
--------------------------------------------------------------------------------------------------
add_14 (Add)                     (None, 7, 7, 2048)   0           bn5a_branch2c[0][0]
                                                                  bn5a_branch1[0][0]
--------------------------------------------------------------------------------------------------
activation_43 (Activation)       (None, 7, 7, 2048)   0           add_14[0][0]
--------------------------------------------------------------------------------------------------
res5b_branch2a (Conv2D)          (None, 7, 7, 512)    1049088     activation_43[0][0]
--------------------------------------------------------------------------------------------------
bn5b_branch2a (BatchNormalizati  (None, 7, 7, 512)    2048        res5b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_44 (Activation)       (None, 7, 7, 512)    0           bn5b_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res5b_branch2b (Conv2D)          (None, 7, 7, 512)    2359808     activation_44[0][0]
--------------------------------------------------------------------------------------------------
bn5b_branch2b (BatchNormalizati  (None, 7, 7, 512)    2048        res5b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_45 (Activation)       (None, 7, 7, 512)    0           bn5b_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res5b_branch2c (Conv2D)          (None, 7, 7, 2048)   1050624     activation_45[0][0]
--------------------------------------------------------------------------------------------------
bn5b_branch2c (BatchNormalizati  (None, 7, 7, 2048)   8192        res5b_branch2c[0][0]
--------------------------------------------------------------------------------------------------
add_15 (Add)                     (None, 7, 7, 2048)   0           bn5b_branch2c[0][0]
                                                                  activation_43[0][0]
--------------------------------------------------------------------------------------------------
activation_46 (Activation)       (None, 7, 7, 2048)   0           add_15[0][0]
--------------------------------------------------------------------------------------------------
res5c_branch2a (Conv2D)          (None, 7, 7, 512)    1049088     activation_46[0][0]
--------------------------------------------------------------------------------------------------
bn5c_branch2a (BatchNormalizati  (None, 7, 7, 512)    2048        res5c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
activation_47 (Activation)       (None, 7, 7, 512)    0           bn5c_branch2a[0][0]
--------------------------------------------------------------------------------------------------
res5c_branch2b (Conv2D)          (None, 7, 7, 512)    2359808     activation_47[0][0]
--------------------------------------------------------------------------------------------------
bn5c_branch2b (BatchNormalizati  (None, 7, 7, 512)    2048        res5c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
activation_48 (Activation)       (None, 7, 7, 512)    0           bn5c_branch2b[0][0]
--------------------------------------------------------------------------------------------------
res5c_branch2c (Conv2D)          (None, 7, 7, 2048)   1050624     activation_48[0][0]
```

```
____________________________________________________________________________________________________
bn5c_branch2c (BatchNormalizati (None, 7, 7, 2048)   8192        res5c_branch2c[0][0]
____________________________________________________________________________________________________
add_16 (Add)                    (None, 7, 7, 2048)   0           bn5c_branch2c[0][0]
                                                                 activation_46[0][0]
____________________________________________________________________________________________________
activation_49 (Activation)      (None, 7, 7, 2048)   0           add_16[0][0]
____________________________________________________________________________________________________
conv2d_1 (Conv2D)               (None, 2, 2, 256)    8388864     activation_49[0][0]
____________________________________________________________________________________________________
reshape_1 (Reshape)             (None, 128, 8)       0           conv2d_1[0][0]
____________________________________________________________________________________________________
lambda_1 (Lambda)               (None, 128, 8)       0           reshape_1[0][0]
____________________________________________________________________________________________________
dropout_1 (Dropout)             (None, 128, 8)       0           lambda_1[0][0]
____________________________________________________________________________________________________
digitcaps (CapsuleLayer)        (None, 8, 16)        132096      dropout_1[0][0]
____________________________________________________________________________________________________
out_caps (Length)               (None, 8)            0           digitcaps[0][0]
====================================================================================================
Total params: 32,108,672
Trainable params: 8,519,936
Non-trainable params: 23,588,736
____________________________________________________________________________________________________
```

Listing 4. CustomNet Model Summary