\title{Modular network for object detection deep neural network optimization }

\begin{abstract}

We present a novel modular convolutional neural network that significantly improves the accuracy of object detection. The network consists of two stages in a hierarchical structure. The first stage is one network that detects general classes. The second stage consists of separate networks to refine the classification and localization of each of the general class objects. Compared to a state-of-the-art object detection network, the classification error in the modular network is improved by approximately 3-5 times, from 12% to 2.5%-4.5%. This network is easy to implement and has a 0.94 mAP. The network architecture is a platform to improve the accuracy of widespread state-of-the-art object detection networks and other types of deep learning networks.

We show that a deep learning network initialized by transfer learning becomes more accurate as the number of classes it trains becomes smaller.

\end{abstract}

\section{Introduction}

In this paper, we present a novel, highly accurate deep learning network for computer vision object detection, in particular, for fine-grained object detection. There is constant effort to increase the accuracy of deep learning networks for object detection. A major topic in object detection is fine-grained detection for distinguishing differences between similar object classes .

The main principles that guide the building of our network are modularity and hierarchy. Our object detection network is denoted as a modular network, consisting of two stages. The first stage is one object detection network for detecting multi-class objects where the classes are general. The second stage consists of separate object detection networks, each trained to detect only similar and related classes that belong to one of the general classes of the first stage network.

Images in the first stage of detected objects belonging to one of the general classes are passed on to the appropriate network in the second stage for detailed identification of an object's type and location. We compared the detection results of our modular network to a state-of-the-art multi-class object detection network, which was trained to detect the same classes as the modular network. The experiments showed that our modular network has significantly higher accuracy.

Our contributions in this paper are: 1) a simple to implement a highly accurate, modular, and hierarchical network for fine-grained object detection. We show both experimentally and theoretically that a deep learning network designed to detect a small number of classes and initially trained by transfer learning is more accurate than a network trained on more classes.

The modular network architecture suggested in this paper can be used to increase the accuracy of state-of-the-art object detection networks by integrating them as parts of the building blocks of this network without changing the intensive optimizations carried out on them. Other types of networks can improve their accuracy by being inserted into this modular network platform.

\section{Related Work}

\subsection{Object detection}

Notable convolutional neural networks for object detection are \citep{journals/corr/Girshick15,journals/corr/LiuAESR15,DBLP:journals/corr/RedmonDGF15,43022}. Faster R-CNN \citep{DBLP:journals/corr/RenHG015} consists of a classification network, a region proposal network which divides the image into rectangular regions, followed by regression for additional accuracy in classification and location. Most of the state-of-the-art object detection networks include a core image classification network, such as Alexnet \citep{Krizhevsky2012ImageNetCW}, VGG \citep{simonyan2014very} or Resnet \citep{DBLP:journals/corr/HeZRS15}. These networks use transfer learning based on the training on a large image data, set such as Imagenet \citep{10.1007/s11263-015-0816-y} and Coco \citep{lin2014microsoft}.

\subsection{Hierarchical structures} Hierarchical structures appear in many forms in computer vision. Fukushima \citep{DBLP:journals/nn/Fukushima88} and Jarrett et al. \citep{inproceedings} proposed a neural network for visual pattern recognition based on a hierarchical network.

\section{The modular network}

\subsection{Modular network architecture}

We present in this paper a new modular and hierarchical object detection network. The network consists of two stages. The first is a deep learning, object detection network trained to detect predetermined general classes, and the second stage consists of several, deep learning, object detection networks, each trained on more fine-grained classes belonging to the same single general class of the first stage network. All the building block networks inside the modular network are trained on negative images as well.

Each independent, deep learning network in the modular network independently goes through the complete object detection process of training and inference. The full input image data set for inference is inserted into the first stage network. If an object in an image is detected as belonging to one of these network classes, the image is passed onto inference by the second stage network trained to detect sub-classes of this class. The purpose of the second stage network is to distinguish between objects of similar classes making more detailed classification and more accurate locations of each object in the image. Each sub-network in the modular network is initialized by transfer learning weights \citep{DBLP:journals/corr/HuhAE16,Karpathy\_2014\_CVPR,oquab2014learning,IEEE\_Transactions,DBLP:journals/corr/YosinskiCBL14} trained on the ImageNet database. Figure 1 shows the modular network of our experiment. The building blocks of the modular network create a Faster-RCNN network \citep{DBLP:journals/corr/RenHG015}. In the first stage, there is a single network trained to detect five general classes of a class object detected in an inference image. With no changes from when it entered the first stage network, this image is passed onto fine-grained detection in the second stage at the appropriate network trained to detect detailed classes belonging to the general class detected in the first stage.

\begin{figure}[h!]

\centering

\includegraphics[width=0.65\textwidth,height=0.25\textheight]{net.jpg}

\label{graph3}

\caption{A modular network whose first stage is a single deep learning network trained to detect five general classes. Its second stage network consists of five separate networks, each trained to detect two distinct sub-classes of one of the general classes.}

\end{figure}

One of the main reasons why the building blocks of our modular networks make the whole network more accurate than a regular multi-class network is that each of the building block networks inside our modular network is designated to detect fewer classes than a regular multi-class network. A possible further modification of the modular network is to add more than two hierarchical stages.

\subsection{Algorithm and deep learning network construction}

To detect multiple classes, we use an object detection network trained by transfer learning. It merges classes of similar labels into a general class label. This network, denoted as the first stage network, is trained to detect new general classes $C\_i$ and additional negative images with no labels that do not belong to any of these general classes. For each of the general classes $C\_i$, we train a second stage network on the same images used to train the detection of the general class and on negative images. This time, we sort and label the training images with fine-grained classes all belonging to this general class. It is possible to train the network on other images with objects belonging to these fine-grained classes.

Images are then input into the first stage network for inference. Images with objects detected as belonging to a general class are passed onto the second stage network dedicated to that class. Those images are then input into the appropriate second stage network for fine-grained object classification and location.

\subsection{Advantages and risks of the modular network}

In each of the sub-convolutional neural networks inside the modular network, there are fewer classes than in a regular network designated to detect the same number of classes as the whole modular network. Thus, there are more features, filters, and network parameters dedicated to the detection of each class, resulting in better accuracy. A small number of features allows less distinction in detection of similar classes as well as errors in detecting rare class objects. When the number of features is small, more features are formed to identify objects types that appear in many images in the training and to detect images of multiple classes, adding errors in fine-grained object detection.

Fewer classes in the object detection network means potentially fewer bounding boxes of detected objects in the image, which gives fewer errors in identifying the objects and finding their locations.

In the modular network training, there are fewer images in the input data set for each of the second stage because the training images are distributed over several networks. This results in fewer parameters and dilution of the features in each image or object by images and objects that do not belong to the classes designated for object detection.

The advantage of the modular network compared to detection by networks that detect many classes at once which have little or no connection to each other is that the hierarchical structure drastically cuts down the number of required inferences, as they are arranged in a tree structure.

The accuracy of the modular network will be better than a multi class network when:

\begin{equation}

a < (a+\Delta\_{1})(a+\Delta\_{2})

\end{equation}*a* represents the multi-label network accuracy; $\Delta\_{1}$ is the improvement in accuracy of the first stage of the modular network compared to the multi-class network; and $\Delta\_2$ shows the improvement in accuracy of the second stage compared to the multi-class network.

Assuming we use the same type of object detection network as the multi class network as the building block network of the modular networks. If the multi-class network has low accuracy, then it is preferred since the building blocks networks inside the modular network should have a very large improvement in accuracy compared to the multi-class network. Most state-of-the-art object detection networks are accurate enough to use them as the building block network for the modular network allowing a modular network with higher accuracy compared to the selected state-of-the-art object detection network. A risk of the modular network is the detection of false negatives in the first stage network. This may reduce accuracy, as some images with true objects may be omitted from the input of the second stage network. To deal with this problem, we designed a second version of the modular network specifically for the image sequence where one object is assumed to appear in more than one image. The network architecture of this version, denoted as Modular Network v.2 is the same as Modular Network v.1, but the difference is that all of the image sequence are sent for inference to the networks. In the second stage, the fine-grained classes are matched to the general classes of the objects detected in the first stage. In this way, the loss of accuracy due to false negative detection in the first stage is reduced.\\ \\

\section{Convolutional neural network classification error model.}

This model describes how reducing the number of classes for detection in a convolutional neural network (CNN) reduces the network classification error. Each of the building block networks inside the modular network has fewer classes than the regular multi-class network. Let x= \{ $x\_1$…$x\_f$\} be the features space. Let *c* be a set of classes c=\{$c\_0$…$c\_n$\}. Every detection of an object in an image is defined by a set of features that are active if this object appears in the image. For example, the features set \{$x\_m$…$x\_p$\} identifies objects belonging to class $C\_1$. *N* represents the total number of features of the designated classes that the CNN can identify. *L* and *T* are the numbers of features of the designated classes that the CNN can identify based on transfer learning and fine tuning \citep{DBLP:journals/corr/YosinskiCBL14}, respectively, where each feature belongs to a single class. *U* is the number of features that the CNN can identify that are common to several classes. N= L+T+U. When each of the designated classes has a similar number of training images, *$S$* - the number of features detecting a designated class, is $S\approx $ $\frac{N}{n}$ $\approx\frac{{L+T}}{n}+U$. In this approximation, the number of features for detecting a single designated class is inversely related to *n*, the number of the CNN designated classes. The smaller the *n*, the more features there are for detecting the class designation, making this object class detection more accurate. The parameters that determine *K* - the number of features that a CNN can identify include: *r* - the number of parameters in the CNN; *a* – the number of filters; *d* – the size of filter; *h* – the number of filter channels; and *q* – the number of layers in the CNN. These parameters are constant for each network. In this model, every CNN has an upper bound with the total number of features, \textit{ sup} *K(r,a,d,h,q)*, that it can identify without increasing the classification errors. Classification error caused by having more features than the optimal number for the network can be, for example, from two channels in the same filter where the weight patterns formed in each channel detect features of different classes. The two patterns can have partial overlap in shape and location. *M* and *B* are output matrices of the convolution of each channel with the corresponding feature map channel. If there is a feature in Matrix M, part of this feature can appear in Matrix B, too, and in the following:

$\sum{\_{{i,j\in G}} ({|M|}\_{i,j}+|B|}\_{i,j}})>{|M|}\_{i,j}$

*G* is a set of all the *i,j* pairs, where *i* and *j* have the respective values of the raw and column indices of pixels included in this feature area. This result is deformation of a feature in the filter's feature map which can cause classification error in the sum of all the feature map channels.

We use Bayes error to estimate the classification error \citep{article1,article2,568732,biobayes}. As an example, we analyzed the classification of two fine-grained classes, $C\_1$ and $C\_0$. According to Bayes error estimation, when there is a probability density where a feature $x\_i$ is activated, i.e, there is a probability that Feature $x\_i$ appears in the feature map when there is an object of class $C\_0$. There is also a probability density that feature $x\_i$ is activated when an object of class $C\_1$ is in the image. The classification error caused by feature $x\_i$ is the smallest probability density between these two probabilities densities. The sum of the smallest probability densities for classification errors of all the features is the classification error. Assuming the probability densities to be activated by classes $C\_1$ or $C\_0$ are known for each of the features in the network, the probability for classification error is described in Equation 2, where P($C\_0$) and P($C\_1$) are the prior probability densities of classes $C\_0$ and $C\_1$, respectively. P($x\_i|$ $C\_0$) and P($x\_i|$ $C\_1$) are the conditional probability densities that feature $x\_i$ is active if the class is $C\_0$ or $C\_1$, respectively. An additional criterion in Equation 2 is the significance of the feature $x\_i$ in the classification. The weights for classes $C\_0$ and $C\_1$ are denoted by $w\_i(C\_0)$ and $w\_i(C\_1$), respectively. If an active feature does not influence the classification of an object, it does not contribute to the classification probability of the object class. The criterion weights $w\_i(C\_0)$ and $w\_i(C\_1$ ) are based on how many times feature $x\_i$ is essential for classification out of all the times this feature was activated by its class objects.

\begin{equation}

P\_{error}=\sum\_{i=1}^{N\_f}min(P(x\_i|C\_0)P(C\_0)w\_i(C\_0), P(x\_i|C\_1)P(C\_1)w\_i(C\_1))\end{equation}

The probability densities of the features are presented in discrete values, which we approximate as a continues graph.

In Graphs 1 and 2, the X-axis is the feature range, denoted as $N\_f$. The Y-axis presents the probability density that a feature is activated. In the graph, all features with the probability of matching a particular class are in the same area on the X-axis. Features with a probability of matching two classes are displayed in the graph in a shared area for both classes. The Bayes classification error is the sum, or integration, of the minimal probability densities of every feature within the mutual area, which is the overlapping of the class $C\_0$ and $C\_1$ curves.

\begin{figure}[h!]

\caption{Graph.1. Ten classes network. The x-axis is the features range denoted as Nf. The Y axis is the features probably densities denoted as f(xi). Graph.2 is a two classes network}

\end{figure}

Graph 1 illustrates the probability densities of identifying features $C\_0$ and $C\_1$ in a network trained to detect ten different classes. The active features are nearly a quarter of the total features in the network. The area of misclassified features is significant compared to the total areas of feature classes $C\_0$ and $C\_1$ which indicates a large classification error. This is because there are many classes and the number of features dedicated to each class is small, which results in a shortage of features necessary to identify the fine-grained features. Since there are many classes, the total number of features exceeds the supermum number of filters for this network resulting false detections.

Graph 2 illustrates a network trained to detect only two classes and negative images. Most of the features detected by this network are of classes $C\_0$ and $C\_1$. The misclassified feature area is small compared to the total area of both classes, indicating that the classification error is small. The number of features for each class is large enabling the training of features for detecting even more detailed features, further reducing the classification error.

\section{Experiments}

\subsection {Implementation }

The original data set for training contains 522 images distributed between 10 classes for five pairs of similar classes. The images, expanded to 46,044 training examples by mirroring, sharpness, brightness, and contrast augmentations, are used as the training data by both the modular and the multi-class networks.

The size of each of the original images in the data is up to 800x800 pixels. The size of the output images of the network is also 800x800 pixels. For the multi-class network and the building blocks of the modular network, we used the state-of-the-art object detection network, Faster R-CNN, with a backbone classification network, VGG 16. The Faster R-CNN network is initialized by training on the ImageNet 2012 database containing 1.2 million images for training and 50k validation images over 1,000 categories. To compare between the multi-class network and the modular network, the sub-networks inside the two networks all have the same hyper-parameter values previously optimized on classes other than those the networks are training to detect. Finely tuned training occured in all the networks inside both the modular network and the multi-class network and included all the networks layers. Each of the networks trained for 40 epochs, with learning rates of 0.001 on the first 10 epochs; 0.0001 on the next 10 epochs; and 0.00001 on the last 20 epochs. The test data set contained 125 original images distributed similarly between four classes: two dog species, Pekinese and Spaniel, and two planets, Mars and Saturn. Both the modular and the multi-class networks inferred on this test data. Most of the original images for the training and the test sets were taken from the Caltech 101 image database and the rest were randomly chosen from the internet.

\subsection {Experiments results }

\subsubsection{Multi-class network}

The multi-class object detection network was trained to detect ten classes and negative images, with a training loss of 0.0229 for Faster RCNN, as defined in \citep{DBLP:journals/corr/RenHG015}. The multiclass network inference results are 0.87 mAP with a 12% error.

\subsubsection{Modular network}

The modular network has two stages. The first stage network is trained on the same data set as the multi-class network including the negative images, but it is labeled with five general classes instead of the more detailed 10 classes of the multi-class network. The modular network’s first stage classes are *dog*, *planet*, *bike*, *boat*, and *bird*. Each of these classes is a unification of a couple of similar classes from the 10 classes labeled for training by the multi-class network. The training loss is 0.0216.

In the second stage, each network is trained on two fine-grained or similar classes and the same negative images as the multi-class network. For example, one network trains on two dog species classes, *Pekinese* and *Spaniel*, with a training loss of 0.0151 loss, while a second network is trained to detect two planets, *Mars* and *Saturn*, with a training loss of 0.0170. The network trains only on images of these classes from the initial training data set.

The modular network v.1 inference results are 0.94 mAP with 4.5% error. The modular network v.2 inference results are 0.95 mAP with 2.5% error. \\

The experimental results indicate that the modular network is significantly more accurate than the multi-class network.

\begin{table}[h!]

The modular network v.1 AP is calculated by taking into account the images detected as false negatives in the first stage of the modular network and thereby not appearing on the mAP of the second stage. Each false negative precision is rated as zero, and its counterpart in the calculation of the whole modular network mAP is one divided by the total number of inference images in the modular network. For example, in Table 1, the AP of *Saturn* in the modular network v.1 is 0.91, but the AP of *Saturn* in the second stage network is 0.94.

Table \ref{percent} shows the experimental results of the network classification errors. The modular network error is significantly reduced to 6\% and 3\% error for *dogs* and *planets* compared to 14\% and 10\%, respectively, in the multi-class network.

\begin{table}[h!]

\begin{figure}[h!]

In the in the first three rows of the first column of Figure 5, the images detected by the multi-class network have errors in classification. However, the general class network and the fine-grained network both detected the same objects correctly.

The second row of images shows that the detection of object location is more accurate in the image detected by the fine-grained network (right) compared to the image detected by the multi-class network (left).

\section{Discussion}

Our results show that most of the classification errors in the multi-class network are between similar classes. The accuracy of the modular network for both v.1 and v.2 is higher by 7.5\%and 9.5\%, respectively, compared to the multi-class network. This is a reduction of classification error by 2.7 and 4.8 times, respectively. The training results indicate that the training loss becomes smaller as the number of classes trained to be detected by the network becomes smaller. The classification error in the modular network is smaller for the *planet* class than the *dog* class, meaning that the two planet classes are less similar to each other than the two *dog* classes. The classification error is smaller when the fine-grained sub-classes are more distinct.

A fundamental question in machine learning is what kind of learning has higher accuracy - a network trained to detect only few focused classes or one that is trained to detect many classes of a wide range of subjects. We ascertain that a network initially trained on a wide range of classes by transfer learning and later trained to detect fewer classes by fine tuning on all the network layers is more accurate than a network initialized by transfer learning and later trained to detect increasingly larger numbers of classes. Previous works on transfer learning \citep{DBLP:journals/corr/HuhAE16,DBLP:journals/corr/YosinskiCBL14} determined that a network initially trained by transfer learning and later trained to detect the designated classes is more accurate compared to only being trained to detect the designated classes. From both findings, we conclude that a network initially trained by transfer learning and then designated to detect a small number of classes is more accurate than if it were designated to detect a larger number of classes.

\section{Conclusion}

The modular network presented in this paper significantly improves object detection performances in both classification and location. This is true especially for detection that requires differentiating between similar classes. This modular network improves state-of-the-art deep learning object detection networks without requiring changes in networks architecture or hyper-parameters. We found that reducing the number of classes a convolutional neural network is trained to detect increases the network accuracy. This modular network could be a platform for other types of deep learning networks, for example, segmentation, by improving their accuracy by implementing them as building blocks of a modular network. This modular network can be applied for fine-grained pattern recognition in artificial intelligence, medical image detection, and scientific research.