# Exploring Better Food Detection via Transfer Learning

anonymous

## Abstract

*In this paper, we present a food-specialized detection[1] deep learning architecture with knowledge transferred from a pretrained food/non-food classification model. Existing approaches in object detection all separate it from image classification due to their incompatible outputs, whereas our work bridges the gap between the two most fundamental computer vision topics by making use of transferred features, and as such we contend that our work provides a new perspective in object detection. Experiments are conducted in two parts. First, transfer learning quantification experiments show that initializing a network with transferred features from classification task can surprisingly produce a boost to generalization even for the detection task. Second, experiments on three state-of-the-art neural networks as backbone show that our approach enables rapid progress and improved performance. The results significantly surpass all original plain networks with more than 10% precision improvement. In addition, our scheme can be easily generalized to any CNN-based architecture.*

## 1 Introduction

The adherence to dietary self-monitoring plays an increasingly important role in public health [1]. For convenience reasons, self food monitoring has been progressively automated, and conducted using image-based food detection. In fact, through projects such as GoCarb [2], the automated food recognition is considered as a "holy grail" of nutrition tracking. Due to the widespread use of imaging devices like digital camera on smart phones, food detection has become a practical technique that could be applied in real-time.

Deep Convolutional Neural Networks (DCNN) [3] brought breakthroughs when applied to cognitive tasks related to video, images, and texts [4]. Existing object detection methods fall into two main categories [5]. First, *two-stage* detection framework which includes a pre-processing step for region proposals, e.g. Fast R-CNN[6]. It initially generates category-independent region proposal, and then category-specific classifiers are used to determine the category labels of region proposals. Second, *one-stage* detection framework, e.g. YOLO9000[7], use fewer computation resources by directly predicting class probability and bounding box, thereby avoiding region proposal generation.

Our work is motivated by the fact that, despite dramatically improved performance of the existing state-of-the-art object detection approaches, food detection has not achieved satisfactory results. The challenges stem from two causes: the vast range of intraclass variations, and the huge amount of categories. On the other hand, thanks to CNN, category-based image classification is much easier to handle; although, all existing frameworks separate detection task from image classification. Under this context, our work focuses on bridging the gap between classification and detection techniques. We make use of the extracted feature information from a trained food/non-food classifier, and then generalize it to our proposed food detection network by transferring exhorted features. Our experiment results show a significantly improvement, and surpass all plain networks trained from scratch. Moreover, our work is conceptually intuitive and easy to transplant to any CNN backed architecture with the purpose of tackling a specialized category detection task.

This paper contributes as follows. First, we propose a transfer-learning-aided approach to improve the detection performance, as well reduce the training time and faster convergence for a specialized category detection task merging with YOLO9000. Second, we demonstrate the wide applicability by conducting experiments with three different neural network backbones. Third, by experimentally quantifying how well the features transfer from classification task to detection task, we instill confidence in our scheme. To our best knowledge, our approach is the first *one-stage* pipeline in food-specialized detection with transferring features.

## 2 Related Work

**Food Localization.** The problem of food localization has been typically addressed as a binary classification problem by using a neural network model to distinguish whether a given image contains food or not. Image segmentation is also mentioned several times in [8, 9], where authors extract segments using hierarchical segmentation, and then apply joint food recognition through multi-class SVM or DCNN. The main deficiency of these approaches is that they all require either the assistance of additional information or environmental context, such as restaurant data, where the picture was taken, or a pre-built database. Our work aims to tackle food detection problem with only image input, as the real-time image analysis sometimes have

[1] In our paper, we denote **detection** as the effect of both recognize and localize multi-food objects in an image; **food classifier** as a binary classification for discriminating food or non-food image.

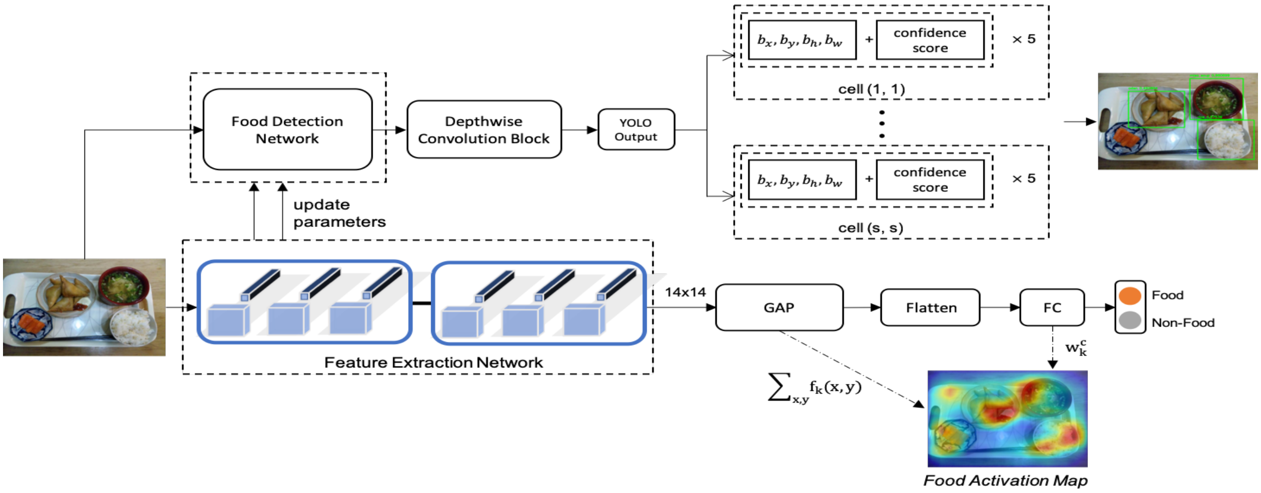


Figure 1: General scheme for our food detection proposal. The bottom part is the pretrained transfer learning process for a binary food classifier, in which we remove last two depthwise convolution blocks to calculate activation maps. The top part is our food detection network with parameters updating from the food classifier, merging with YOLO output layer.

no access to any convenient information.

Bolanos *et. al* [10] proposed the first food localization and recognition approach with two main steps: one is a food localization network built on GoogleNet [11], which can generate bounding box proposals by producing activation maps on the input image and second, recognize each food-related object presents in each bounding box by fine-tuning the GoogleNet used in the first step. However, this approach is a two-stage pipeline that requires much computation. Our proposed approach is built on a one-stage framework, which is computationally less expensive. Moreover, this approach can be efficiently improved by integrating with YOLO, which is one of the state-of-the-art *unified* detection strategies, leading to generation of bounding box and class label simultaneously, rather than separating food localization and recognition into two stages.

**Transfer Learning with Neural Networks.** Object detection is one of the fundamental problems of computer vision. Although some productive and successful solutions exist, they require a massive amount of training data, tedious annotation and long training process. Transfer learning is a deep learning technique built upon previous feature information extracted from other pre-trained neural network model, thereby reducing training requirements. In our work we address how to benefit a food detection task from food/non-food classification via transfer learning, so that our food detection approach has a superior capacity of learning based on previous experiences and knowledge.

Although there exist many image classification papers performing transfer learning [12, 13], research in object detection, especially when it comes to food, is rather limited. Rajpura *et.al* [14] presented a food detection approach with transfer learning to detect packaged food products in refrigerator scene by manually generating synthetic images as the training set. However, their model can only find the approximate position of food packages, but not a food class or confidence. Further, the process of hand-crafting datasets is not only tedious, but works only for a specific scenario.

## 3 Methodology

In this section, we describe the proposed methodology in two steps: a) training a binary food classifier on Food-5K, b) training a food detection network on UECFood100 and UECFood256 with transferring features followed by a YOLO output layer. The whole process is illustrated in Fig. 1.

### 3.1 Food/Non-Food Classifier

Our food-specialized detection algorithm starts with training a binary food classifier to discriminate whether input images contain food. Then, we calculate Food Activation Maps (FAM) [10] to highlight regions that are most relevant to food. In other words, we pre-train a binary classifier to sensitize our model to real food regions, as shown by a FAM example in the lower part of Fig. 1.

**Food Activation Maps Generation.** Once we finish the training of the food classifier we calculate FAM. In order to do that we remove the last two depthwise convolution blocks to get a 14x14 output for sufficient spatial resolution, and then we connect with a Google Average Pooling(GAP) layer [15] to calculate FAM. Finally, we fine-tune our model on Food-5K dataset again, such that it is suited for further transfer learning.

By applying the output of GAP into the class score, $S_c$, we obtain

$$S_c = \sum_{x,y}\sum_k w_k^c f_k(x,y) = \sum_{x,y} FAM_c(x,y) \quad (1)$$

where $FAM_c(x,y)$ indicates the importance of the activation at spatial grid $(x,y)$ containing a food item. $c \in \{0,1\}$ represents binary categories, and $w_k^c$ is the weight corresponding to the category for unit $k$.

Despite the work [10] showing that it is feasible to extract bounding boxes through heatmap segmentation, we do not rely on FAM to extract food objects because this approach requires high resolution input to clearly separate the food objects, as well as massive amounts of fine-tuning on various datasets. Otherwise the FAM could not tell the edges of different food objects, as illustrated in Fig. 2.

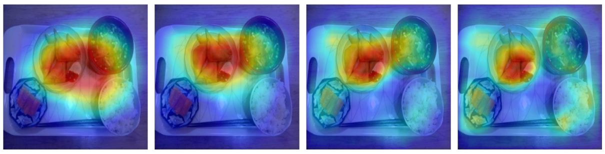


Figure 2: A disadvantage of using FAM to perform localization. The input image resolution is (299, 299), (399, 399), (499, 499) and (599, 599) from left to right.

At the end of this stage we acquire three food/non-food classifiers corresponding to three backbone neural networks with 95% average accuracy.

### 3.2 YOLO Output Layer

We now describe the final layer as illustrated in the top right corner of Fig. 1. Our food detection network has the same structure as the feature extraction network. However, it is followed with two more depthwise convolution blocks [16]. This helps to downsize the resolution to $(7,7)$, dividing the image into a $7 \times 7$ grid. Finally, a YOLO layer follows to reshape the output. We use a k-nearest neighbours (KNN) algorithm to calculate 5 anchor boxes. Each grid cell predicts 5 bounding boxes to locate the object. Each bounding box comprise of 5 elements: $b_x, b_y, b_w, b_h$ and a box confidence score, which reflects the likelihood of a box containing an object. Hence, the final output box after YOLO layer has a shape of (S, S, #box, 5 + #class) = (7, 7, 5, 5 + #class).

The confidence score is calculated by the following equation:

$$CS = P_r(class_i) \cdot IoU_{pred}^{truth} \quad (2)$$

where $P_r(class_i)$ is the probability that the object belongs to $class_i$. $IoU_{pred}^{truth}$ is the Intersection over Union (IoU) ratio between the predicted box and the ground truth.

### 3.3 Transfer Learning for Food Detection

The final step of our food detection network is to fine-tune the food detection datasets (UECFood100[17] and UECFood256[18]) using transferring features from pre-trained food classifier.

In Fig. 1 we use MobileNet [19] as the example feature extraction network to illustrate how to apply our transfer learning scheme for food detection. Further, we prove the efficiency and applicability of our solution on three different neural networks in Sec.4.2. MobileNet is built on depthwise separable convolution blocks [16], which are factorized forms that express a standard convolution by a depthwise convolution and a 1x1 pointwise convolution. By this two-step process of filtering and combining we get a significant parameter reduction [19].

In the next section we present a number of experiments to quantify the transfer learning effect.

## 4 Experiments

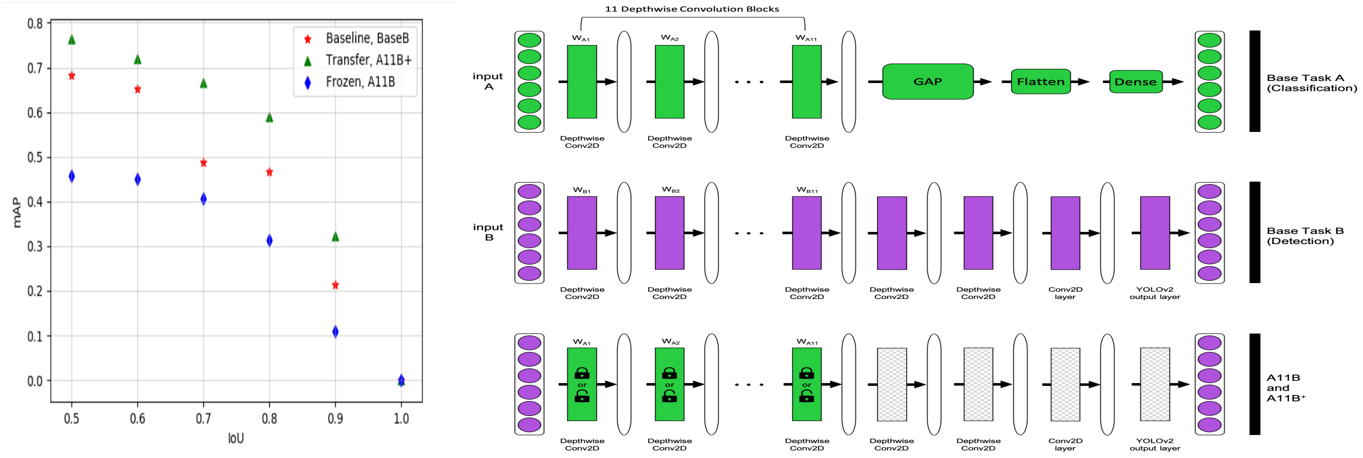
We use three datasets to build and evaluate the proposed approach: Food-5K [20] which contains 50% food images and 50% non-food images for training food/non-food classifier, and UECFood100 [17] and UECFood256 [18] with 100 and 256 kinds of Japanese food with 40K images in total. We design transfer learning quantification experiments in Sec. 4.1 and ablation experiments in Sec. 4.2 on three state-of-the-art neural networks. Finally, we analyze the effect of transferring features and the performance of our transfer-learning-aided (TLA) food detection technique in comparison with all plain neural networks trained from scratch.

### 4.1 Transfer Learning Quantification

Yosinski *et. al* [21] proposed a experimental method to quantify the generality versus specificity of neurons in different layers. We assume that our base model (food classification) contains *general* information for food detection, so features in all CNN layers of the food classifier are being transferred. To prove our premise, we design similar experiments but instead of splitting the same dataset into two parts for task `A` and task `B` for the same type of task, we consider food/non-food classification as task `A` and food detection as task `B`, training with entirely different datasets, as shown in the top two rows of Fig. 3(b). These networks, which we call `baseA` and `baseB`, are modified accordingly on the last several layers to generate distinct outputs for different tasks.

We demonstrate how well features from all convolution blocks transfer from base task to another one by defining and training the following two networks, as illustrated in the bottom row of Fig.3(b):

- A *transfer* network `A11B`: the first 11 depthwise convolution blocks are copied from `baseA` and frozen. The remaining higher layers are initialized randomly, except last several Conv2D layers' weights, which are initialized based on the weight



(a) (b)

Figure 3: (a)Results of transfer learning quantification experiments. (b)Overview of the transfer learning quantification experimental treatments and controls. The labeled rectangles (e.g $W_{B1}$) represent the weight vector learned for that layer, with the color indicating which dataset and task the layer was originally trained on.

shape and normalized by the grid size of YOLO output layer. They are trained for the dataset `B`.

- A *transfer* network `A11B+`: just like `A11B` but layers are not frozen but trainable.

The results of all `A/B` transfer learning experiments on dataset Food-5K (task `A`) and UECFood100 (task `B`) are shown in Fig. 3(a). The red stars are the results of `baseB`, indicating a network trained from scratch for food detection task with 49.4% average mAP from 0.5 IoU to 0.9 IoU. In both `A11B` and `A11B+` experiments, we compare their performance with the `Baseline`.

The blue `A11B` diamonds show the transferability of features from one network of a base binary classification task to food detection task, with 61.25% average mAP. There is a significant drop at IoU 0.5 and 0.6 from `BaseB` to `A11B` meaning that the weights we copied containing *specific* features for task `A` but also *general* features for both task `A` and `B`, thus freezing those features will hurt the performance.

The green `A11B+` triangles show that transferring features from food/non-food classifier benefit the performance of food detection task. Moreover, this result suggests that transferring features will boost generalization performance even if the target task is complex and with a 3 times bigger dataset. In addition, this result is not attributed to longer training time. In fact, we trained all networks with same iterations (12K base iterations vs. 12K iterations for `A11B` vs. 12K finetuning iterations for `A11B+`). The average boost from 0.5 IoU to 0.9 IoU is **11.8%** between `Transfer` and `Baseline`.

### 4.2 Ablation Experiments

To find out the effects of transferring features, we ablated (removed) the transferring features from our food detection model, and then trained with three different neural network backbones to prove applicability and general efficiency. Results are shown in Fig. 4 and discussed in detail text.

**Architecture.** We generalize our approach to three state-of-the-art CNN-based architectures: MobileNet, MobileNetV2 and ResNet18. Please note that we use the same input resolution(224x224) for all these networks.

**Transfer Learning vs. From scratch.** Our food detection approach benefits from pre-training a food/non-food classifier. In Fig. 4 top row, we compare the training loss between our TLA models with the model trained from scratch. We found that all plain backbones' performance have been improved with transferring features. They have a much better initial performance with lower loss. For MobileNet and MobileNetV2, the starting loss is 9 times less and 5 times less respectively, compared with the same backbone model without transfer learning. Further, the same performance is achieved after only half of the number of epochs (red points have the same loss value as the black points, but with 50% less training time), and the overall loss ends up with nearly zero. Even though we do not see a significant difference in the training process of ResNet18, the food detection precision has been greatly improved as shown in Table.1.

**Analysis of mAP vs. IoU.** Next we compute the mean Average Precision (mAP) of output bounding boxes at different IoU ratios with ground truth boxes. Fig. 4 second row shows the mAP result among three backbones with transferring features in comparison with plain networks. The plots illustrate a great improvement in our TLA models illustrated by red lines. Our scheme in all three structures significantly surpasses all plain models with IoU ratios from 0.5 to 1.0. More specifically, there is **11%** mAP improve-

Table 1: **UECFood100** and **UECFood256** detection mean average precision result (%) under three backbone neural networks with or without transferred features. IoU = 0.5.

| Dataset | MobileNet | TLA-MobileNet | MobileNetV2 | TLA-MobileNetV2 | ResNet18 | TLA-ResNet18 |
|---|---|---|---|---|---|---|
| UECFood100 | 68.25 | **76.37** | 59.51 | **78.29** | 53.19 | **61.66** |
| UECFood256 | 69.76 | **75.01** | 73.42 | **76.01** | 40.92 | **54.77** |

ment for MobileNet, **18%** increase for MobileNetV2 and **8%** boost for ResNet18. It achieves nearly **80%** mAP with MobileNet and MobileNetV2. The results show that initializing with transferring features can improve generalization performance even after substantial fine-tuning of a new task, which could be a useful technique for improving object detection performance.

### 4.3 Food Detection Performance

To quantify food detection performance, we choose the most commonly used metric in multi-class object detection problem, mean average precision (mAP), to evaluate our algorithm in the testing set. A positive detection has `IoU` $>$ `0.5` with the ground-truth, `obj_threshold=0.3` to distinguish between non-object and object. Further, `nms_threshold=0.3` is used to determine whether two detections are overlapped and duplicated. To take into account the results of both, localization accuracy and recognition precision, we applied a mean average over all classes. Our proposed transfer-learning-aided architecture is capable of finding most of the food-related objects in both UECFood100 and UECFood256 datasets with the fewest possible bounding boxes.

Finally, in Fig. 5 we show some examples of the complete method. Table. 1 shows the mAP result with `IoU = 0.5` with validation on more than 10K food images of various kinds of food.

## 5 Conclusions and Future Work

We proposed a high precision transfer learning scheme for food detection task. Our approach is applicable to any CNN-based neural networks. The experiments show that that even small networks achieve surprising performance. We have also experimentally quantified how transferability benefits object detection task from image classification. We found that initializing with transferred features can greatly improve generalization performance. Our work is an intuitive combination of image classification and object detection.

In research to follow, we plan to integrate this technique into a comprehensive food tracking solutions that can execute solely on mobile computing devices. Further, the proposed technique has the potential to become useful for improving object detection performance under more general context.

## References

[1] Vinay Bettadapura, Edison Thomaz, Aman Parnami, Gregory D. Abowd, and Irfan A. Essa. Leveraging context to support automated food recognition in restaurants. *CoRR*, abs/1510.02078, 2015. 1

[2] Aristotelis Agianniotis, Marios Anthimopoulos, Elena Daskalaki, Aurlie Drapela, Christoph Stettler, Peter Diem, and Stavroula Mougiakakou. Gocarb in the context of an artificial pancreas. *Journal of Diabetes Science and Technology*, 9(3):549–555, 2015. PMID: 25904142. 1

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 1

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

[5] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikinen. Deep learning for generic object detection: A survey. *arXiv preprint*, 1809.02165, 2018. 1

[6] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 1

[7] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016. 1

[8] M. Anthimopoulos, J. Dehais, P. Diem, and S. Mougiakakou. Segmentation and recognition of multi-food meal images for carbohydrate counting. In *13th IEEE International Conference on BioInformatics and BioEngineering*, pages 1–4, Nov 2013. 1

[9] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp. Food image analysis: Segmentation, identification and weight estimation. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, July 2013. 1

[10] Marc Bolaños and Petia Radeva. Simultaneous food localization and recognition. *CoRR*, abs/1604.07953, 2016. 2, 3

[11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 2

[12] A. Quattoni, M. Collins, and T. Darrell. Transfer learning for image classification with sparse prototype representations. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008. 2

[13] Yin Zhu, Yuqiang Chen, Zhongqi Lu, Sinno Jialin Pan, Gui-Rong Xue, Yong Yu, and Qiang Yang. Heterogeneous transfer learning for image classification. In *Pro-*

*ceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, AAAI'11, pages 1304–1309. AAAI Press, 2011. 2

[14] Param S. Rajpura, Ravi S. Hegde, and Hristo Bojinov. Object detection using deep CNNs trained on synthetic images. *CoRR*, abs/1706.06782, 2017. 2

[15] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2

[16] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016. 3

[17] Y. Matsuda, H. Hoashi, and K. Yanai. Recognition of multiple-food images by detecting candidate regions. In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, 2012. 3

[18] Y. Kawano and K. Yanai. Automatic expansion of a food image dataset leveraging existing categories with domain adaptation. In *Proc. of ECCV Workshop on Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2014. 3

[19] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 3

[20] Ashutosh Singla, Lin Yuan, and Touradj Ebrahimi. Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2Nd International Workshop on Multimedia Assisted Dietary Management*, MADiMa '16, pages 3–11, New York, NY, USA, 2016. ACM. 3

[21] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014. 3

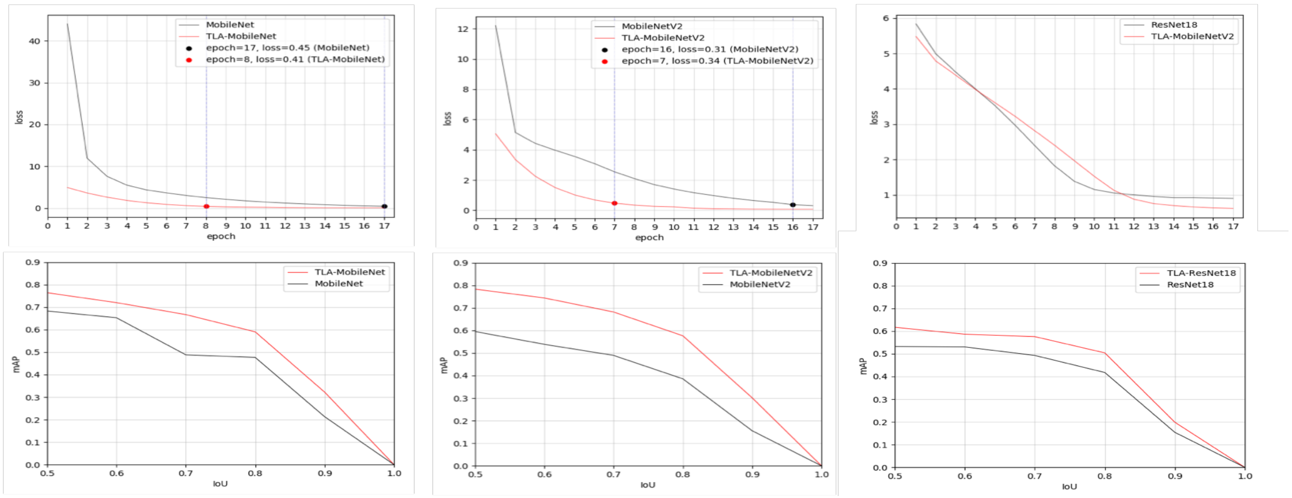


Figure 4: Loss history and evaluation result of ablation experiments on three neural networks. *First row:* Loss history during training with same number of iterations on MobileNet, MobileNetV2, ResNet18. *Bottom row:* mAP results for TLA models and original models at different IoU.

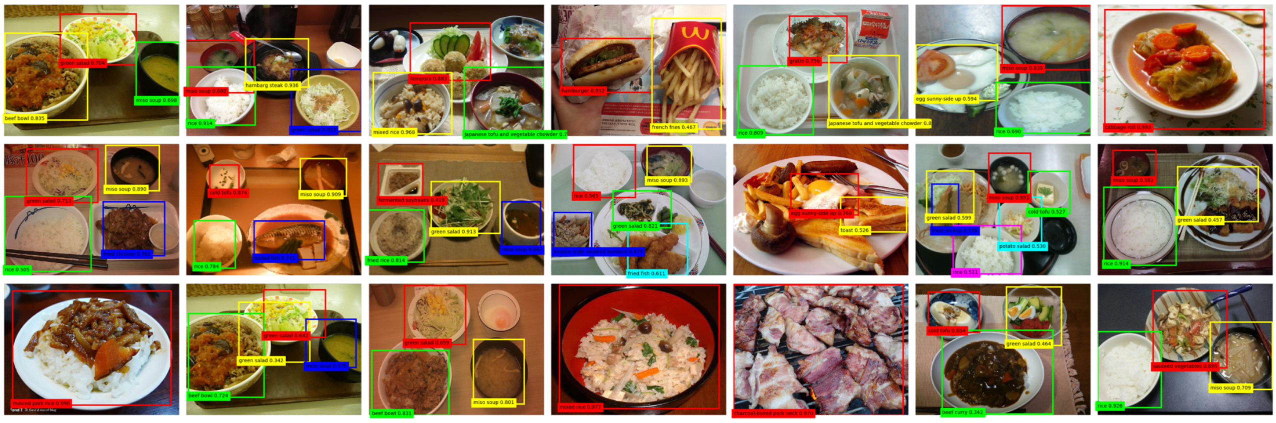


Figure 5: Results of our transfer-learning-aided (TLA) approach on UECFood100 and UECFood256 datasets. The results are based on MobileNetV2, achieving mAP of 78.29%. Bounding box, category and confidence are shown in colors. It is recommended to view this figure in color on-line, to zoom in to observe numbers and labels.