# AI CUP 2020 Mango Image Recognition Challenge: Grade Classification and Defective Classification

## 1. The Mango Grade Sub-Challenge

For the competition of mango image classification, dataset was created at three fruit collection facilities in Fangshan, Pingtung, containing tens of thousands of images. The images were made in well-lighted spaces using smart phones and the video recordings with HDR camera. Each of the image contained a shot of one mango that was put on data collectors' hand or the conveyor. Furthermore, each image was labeled with a mango grade and defective types by professionals. Grade classification includes three grades A, B or C, and specified defective categories. These labels are used as the gold standard of the AI CUP 2020 Mango Image Recognition Challenge.

## 2. Experiments and Results

For the Grade Classification task, dataset was divided into train, development and test sets. The following Table 1 shows the number of samples in each set. In the stage 1 competition, we released train, development and test sets.

Table 1: *Database: Number of image data per class (grade) in the train/development/test splits: Test splits distributions are blinded during the ongoing challenge.*

| Grade \ ML | Train | Dev | Test | Total |
|---|---|---|---|---|
| A | 1792 | 243 | blinded | blinded |
| B | 2068 | 293 | blinded | blinded |
| C | 1740 | 264 | blinded | blinded |
| Total | 5600 | 800 | 1600 | 8000 |

For all of these data, the images were preprocessed before passing into our baseline models. The steps were shown as below:

I. Resize image as (224, 224)
II. Rotate the image by angle (degree = 15)
III. Horizontally flip the image with given probability (p = 0.5)
IV. Normalize the image with mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225]

## 2.1 The Image Feature Extractors

The official baseline feature sets are extracted using a variety of well-known convolutional neural networks which have been used extensively in tasks of image classification, object detection, image pixelwise segmentation and instance segmentation. We selected the following pretrained models as our feature extractor: ResNeXt (ResNeXt-50 32×4d) [1], AlexNet [2], VGG16 [3], DenseNet (Densenet-121) [4] and ShuffleNet (ShuffleNetV2 with 1.0x output channels) [5] (all of them are available on PyTorch[1] pretrained models subpackage[2]). After preprocessing the image, the image tensor was fed into each pretrained model to obtain 1000-dimension image representation.

## 2.2 Grade Classification

For the sake of transparency and reproducibility of baseline competition, we used an open-source implementation of Support Vector Machines (SVM) on scikit-learn[3] with linear kernel and balanced mode. For all tasks, the complexity parameter *C* was optimized during the testing phase. In this Sub-Challenge, we provided three baseline approaches to Grade Classification: Single Net (feature extracted using one specific network), Feature-level Fusion and Decision-level Fusion models. We chose the highest results on the test set as the baseline shown as bold text in Table 2 (we further list the corresponding parameters in Support Vector Machine (SVM) in Table 3).

Table 2: *Results for Sub-Challenge Mango Grade Classification task. The official baselines for test are highlighted (bold and in red). All of accuracy (i.e., weighted averaged recall, WAR) was calculated on* ***test*** *set. ALL: Fusion of all five networks.*

| *Acc(%) on Test Set* | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Single Net** | | | | | | | | | |
| ResNeXt (R) | | AlexNet (A) | | VGG16 (V) | | DenseNet (D) | | ShuffleNet (S) | |
| 59.000 | | 68.938 | | 65.250 | | 49.875 | | 56.125 | |
| **Early-fusion (Feature-level Concatenate)** | | | | | | | | | |
| R+A | 68.938 | A+D | 69.000 | R+A+V | 69.938 | R+D+S | 57.313 | R+A+V+D | 69.375 |
| R+V | 65.250 | A+S | 43.313 | R+A+D | 68.938 | A+V+D | 69.625 | R+A+V+S | 41.750 |
| R+D | 43.438 | V+D | 65.500 | R+A+S | 44.875 | A+V+S | 40.688 | R+A+D+S | 45.250 |
| R+S | 58.750 | V+S | 46.000 | R+V+D | 65.688 | A+D+S | 43.625 | R+V+D+S | 46.688 |
| A+V | 70.188 | D+S | 55.563 | R+V+S | 47.563 | V+D+S | 45.688 | A+V+D+S | 40.750 |
| | | | | | | | | ALL | 42.000 |
| **Late-fusion (Decision-level Concatenate)** | | | | | | | | | |
| R+A | 69.063 | A+D | 69.125 | R+A+V | 70.063 | R+D+S | 56.750 | R+A+V+D | 68.875 |
| R+V | 65.438 | A+S | 55.375 | R+A+D | 69.000 | A+V+D | ***70.250*** | R+A+V+S | 56.813 |
| R+D | 47.750 | V+D | 65.688 | R+A+S | 57.625 | A+V+S | 54.000 | R+A+D+S | 55.750 |
| R+S | 56.500 | V+S | 56.500 | R+V+D | 65.688 | A+D+S | 53.563 | R+V+D+S | 57.500 |
| A+V | 69.625 | D+S | 56.125 | R+V+S | 57.813 | V+D+S | 55.500 | A+V+D+S | 52.313 |
| | | | | | | | | ALL | 56.063 |

Table 3: *Parameters for Sub-Challenge Mango Grade Classification task. C: Regularization parameter in Support Vector Machine. P: Percentage in ANOVA feature selection.*

*Regularization paramters (C) and Percentage (P) on feature selection. (C, P)*

| **Single Net** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ResNeXt (R) | | AlexNet (A) | | VGG16 (V) | | DenseNet (D) | | ShuffleNet (S) | |
| (1, 70) | | (0.1, 30) | | (0.1, 30) | | (1, 100) | | (0.1, 100) | |
| **Early-fusion (Feature-level Concatenate)** | | | | | | | | | |
| R+A | (0.1, 30) | A+D | (0.1, 40) | R+A+V | (1.0, 10) | R+D+S | (0.1, 50) | R+A+V+D | (1.0, 10) |
| R+V | (0.1, 30) | A+S | (1.0, 50) | R+A+D | (0.1, 20) | A+V+D | (1.0, 10) | R+A+V+S | (1.0, 50) |
| R+D | (1.0, 100) | V+D | (0.1, 60) | R+A+S | (1.0, 30) | A+V+S | (1.0, 100) | R+A+D+S | (1.0, 20) |
| R+S | (1.0, 40) | V+S | (1.0, 80) | R+V+D | (0.1, 30) | A+D+S | (1.0, 90) | R+V+D+S | (0.1, 80) |
| A+V | (0.1, 20) | D+S | (0.1, 100) | R+V+S | (1.0, 50) | V+D+S | (0.1, 100) | A+V+D+S | (1.0, 100) |
| | | | | | | | | ALL | (1.0, 100) |
| **Late-fusion (Decision-level Concatenate)** | | | | | | | | | |
| R+A | (1.0, -) | A+D | (1.0, -) | R+A+V | (0.1, -) | R+D+S | (0.1, -) | R+A+V+D | (0.1, -) |
| R+V | (0.1, -) | A+S | (0.1, -) | R+A+D | (0.1, -) | A+V+D | (0.1, -) | R+A+V+S | (1.0, -) |
| R+D | (0.1, -) | V+D | (0.1, -) | R+A+S | (1.0, -) | A+V+S | (1.0, -) | R+A+D+S | (1.0, -) |
| R+S | (0.1, -) | V+S | (0.1, -) | R+V+D | (0.1, -) | A+D+S | (0.1, -) | R+V+D+S | (0.1, -) |
| A+V | (0.1, -) | D+S | (0.1, -) | R+V+S | (1.0, -) | V+D+S | (1.0, -) | A+V+D+S | (1.0, -) |
| | | | | | | | | ALL | (1.0, -) |

**Single Net Classification**

We obtained 1000-dimension feature representation for each of the five pre-trained models. We further performed ANOVA univariate feature selection with a tuning parameter of percentage *P* of features to be selected as input to the classifier. Finally, we show the results of each of these five sets of features on the test set. The criterion for evaluating the model was used as weighted average recall (i.e., accuracy) listed in the Table 2. The feature set that achieves the highest accuracy is the AlexNet (i.e., 68.938%).

### 2.2.1 Fusion Net on Feature-level Concatenate (Early-fusion)

This fusion approach is based on single net, after collecting features from feature selection, in this stage we concatenate the feature sets to obtain the fusion-level features as input to the SVM. As we see in the performance table, there are significant improvement across almost all networks, achieving higher recognition accuracy 70.188%, 69%, 69.938%, 69.625% and 69.375% in A+V, A+D, R+A+V, A+V+D and R+A+V+D models.

[1] PyTorch version: 1.4.0

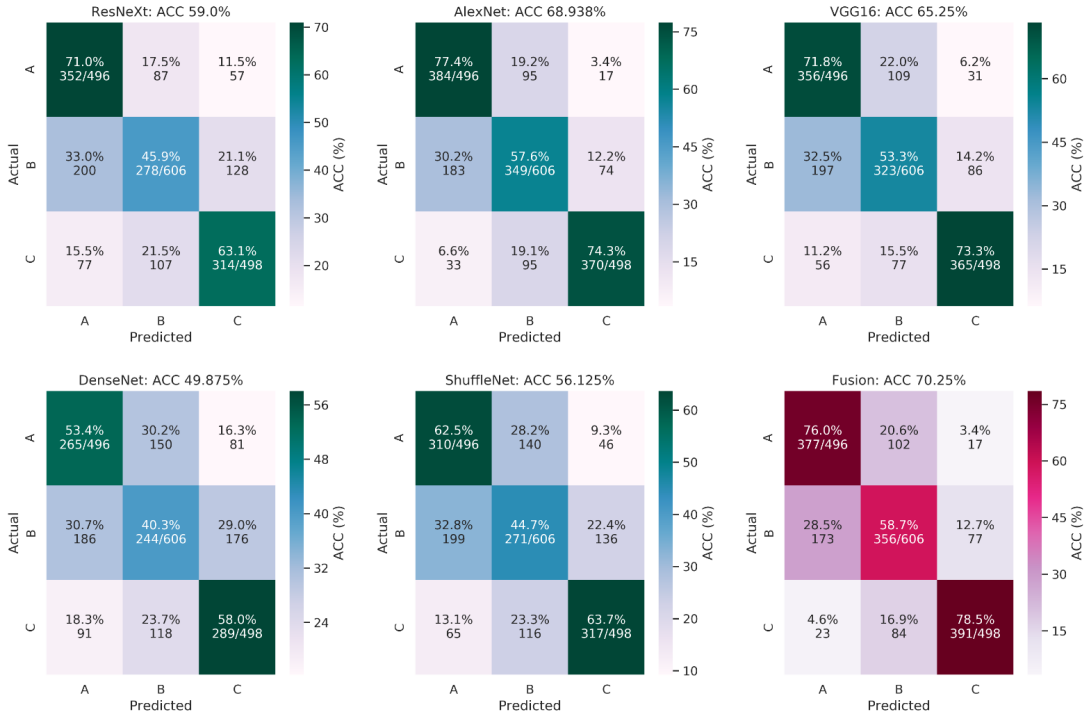
[2] PyTorch pretrained model: https://pytorch.org/docs/stable/torchvision/models.html

[3] scikit-learn version: 0.22.1

### 2.2.3 Fusion Net on Decision-level Concatenate (Late-fusion)

This is a two-stage classification framework. We first perform classification using feature extraction from the Single Net, and we further apply one-versus-rest decision function to obtain the values of distance between sample and the separating hyperplane. These numerical values are used as the feature representation. For the decision-level fusion approach, we achieved the highest recognition accuracy, i.e., 70.25% in 3-class classification task by A+V+D which improves 0.0625% accuracy comparing to the feature-level fusion approach. Consequently, the baseline of Sub-Challenge Grade Classification is **ACC = 70.25%.**

Figure 1: *Confusion matrices on the test set; overall number of classification accuracy given in Table 2. For the Sub-Challenge, we provided confusion matrices of all single net (ResNeXt, AlexNet, VGG16, DenseNet, ShuffleNet) and the best fusion model (Decision-level Fusion on AlexNet+VGG16+DenseNet).*



## 3. References

[1] Xie, Saining, et al. "Aggregated residual transformations for deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[2] Krizhevsky, Alex. "One weird trick for parallelizing convolutional neural networks." arXiv preprint arXiv:1404.5997 (2014).

[3] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[4] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[5] Ma, Ningning, et al. "Shufflenet v2: Practical guidelines for efficient cnn architecture design." Proceedings of the European Conference on Computer Vision (ECCV). 2018.