



## Fruit freshness detection based on multi-task convolutional neural network

Yinsheng Zhang<sup>a</sup>, Xudong Yang<sup>b</sup>, Yongbo Cheng<sup>c</sup>, Xiaojun Wu<sup>d</sup>, Xiulan Sun<sup>e</sup>, Ruiqi Hou<sup>a,\*\*</sup>, Haiyan Wang<sup>a,\*</sup>

<sup>a</sup> Zhejiang Food and Drug Quality & Safety Engineering Research Institute, Zhejiang Gongshang University, Hangzhou, 310018, China

<sup>b</sup> School of Management and E-Business, Zhejiang Gongshang University, Hangzhou, 310018, China

<sup>c</sup> School of Management Science and Engineering, Nanjing University of Finance and Economics, Nanjing, 210023, China

<sup>d</sup> Institute of Science and Technology, Jiangnan University, Wuxi, 214122, China

<sup>e</sup> School of Food Science and Technology, Jiangnan University, Wuxi, 214122, China

### ARTICLE INFO

#### Keywords:

Multi-task learning  
Depthwise separable convolution  
Fruit freshness  
Convolutional neural network

### ABSTRACT

**Background:** Fruit freshness detection by computer vision is essential for many agricultural applications, e.g., automatic harvesting and supply chain monitoring. This paper proposes to use the multi-task learning (MTL) paradigm to build a deep convolutional neural network for fruit freshness detection.

**Results:** We design an MTL model that optimizes the freshness detection ( $T_1$ ) and fruit type classification ( $T_2$ ) tasks in parallel. The model uses a shared CNN (convolutional neural network) subnet and two FC (fully connected) task heads. The shared CNN acts as a feature extraction module and feeds the two task heads with common semantic features. Based on an open fruit image dataset, we conducted a comparative study of MTL and single-task learning (STL) paradigms. The STL models use the same CNN subnet with only one specific task head. In the MTL scenario, the  $T_1$  and  $T_2$  mean accuracies on the test set are 93.24% and 88.66%, respectively. Meanwhile, for STL, the two accuracies are 92.50% and 87.22%. Statistical tests report significant differences between MTL and STL on  $T_1$  and  $T_2$  test accuracies. We further investigated the extracted feature vectors (semantic embeddings) from the two STL models. The vectors have an averaged 0.7 cosine similarity on the entire dataset, with most values lying in the 0.6–0.8 range. This indicates a between-task correlation and justifies the effectiveness of the proposed MTL approach.

**Conclusion:** This study proves that MTL exploits the mutual correlation between two or more relevant tasks and can maximally share their underlying feature extraction process. We envision this approach to be extended to other domains that involve multiple interconnected tasks.

### 1. Introduction

Freshness detection is an important task in the fruit supply chain. The traditional and effective method for assessing fruit freshness is the human sensory evaluation, but this approach is often susceptible to subjective influences from the evaluators (Birwal et al., 2015). As a result, mass spectrometry and chromatographic detection techniques have become a new alternative to traditional ways of freshness detection. Mass spectrometry and chromatographic techniques provide non-destructive detection and are less susceptible to subjective biases from the operators (Ventura-Aguilar et al., 2021). However, due to the high cost of mass spectrometry and chromatographic sensors, some gas sensors have been gradually utilized in practical detection. At the same

time, in certain specific scenarios, specific chemical reagents with particular functions, such as fluorescence probes, are also used for the freshness detection of fruits (Gong et al., 2023). While these methods have achieved certain effectiveness in fruit freshness detection, they all share some common issues, such as high investment costs, low efficiency in processing large-scale data, and reliance on human expertise. Therefore, in recent years, with the development of artificial intelligence, the use of machine learning methods for fruit freshness detection has gradually replaced some traditional detection methods. These methods are known for their non-invasive nature, speed, accuracy, and low cost advantages (Zhong et al., 2023). For example, machine learning methods such as Support Vector Machines (SVM) and decision trees have been used to analyze spectroscopic profiling datasets of fruits to

\* Corresponding author.

\*\* Corresponding author.

E-mail addresses: [houriqi@mail.ustc.edu.cn](mailto:houriqi@mail.ustc.edu.cn) (R. Hou), [whydd@zjgsu.edu.cn](mailto:whydd@zjgsu.edu.cn) (H. Wang).

<https://doi.org/10.1016/j.crfs.2024.100733>

Received 18 January 2024; Received in revised form 20 March 2024; Accepted 4 April 2024

Available online 8 April 2024

2665-9271/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

detect their freshness (Baranowski et al., 2013). Classic deep learning models like GoogLeNet combined with dimensionality reduction techniques have been employed to analyze fruit image data for freshness detection (Yuan and Chen, 2024). For more complex tasks, larger deep learning models like YOLO have been utilized for freshness detection (Salim, 2023). Recently, multi-task learning has shown its potential in various applications, and this paper will provide an alternative multi-task learning (MTL) approach for fruit freshness detection. MTL aims to improve the performance of multiple related tasks by simultaneous training. Related work has proven MTL's effectiveness in various image classification tasks, e.g., face recognition (Lu et al., 2023), medical imaging diagnosis (Zhao et al., 2023; Tang et al., 2023; Sun et al., 2023; Zhou et al., 2021) and food safety (Amrani et al., 2024). However, research on the application of multi-task learning in fruit freshness detection is still underdeveloped and requires further exploration. Thus, this study will explore the possibility of using MTL in fruit freshness detection.

The manuscript is organized as follows. In the method section, we first will design a depth-wise separable convolutional (DSC) network as a common backbone for image classification tasks. Then, we will use the MTL paradigm to design a model that supports two sub-tasks. Finally, we will conduct a comparative study on MTL and STL using an open dataset.

## 2. Method

### 2.1. Backbone model design

The designed neural network contains two parts (Fig. 1). The first part is a shared CNN (convolutional neural network). The second part is a downstream task subnet. The shared CNN aims to extract middle- and high-level features from the images. Its output is a fixed-length latent visual semantic embedding. The downstream task subnet is simply an MLP (multi-layer perceptron), which contains two FC (fully connected) or dense layers. It takes the shared CNN's extracted semantic features as input and outputs the final prediction result.

In the above model, we largely use the depth-wise separable convolutional (DSC) layers. DSC comprises a depth-wise per-channel spatial convolution and a point-wise convolution that combines multiple channels (Fig. 2).

Compared to routine convolution, it has far fewer parameters and significantly low computational cost (Huang et al., 2021; Chollet, 2017). For an input feature map with a size of  $D_k \times D_k \times M$  for the image (input size of  $D_k \times D_k$  and  $M$  channels), the size of the convolutional kernel is

$D_F \times D_F \times M$ , and there are  $N$  kernels in total. Therefore, a single convolution operation requires a total of  $D_k \times D_k \times D_F \times D_F \times M$  computations. For a convolutional layer with  $N$  convolutional kernels, the total computational cost of convolutional layer  $C(\text{routine convolution}) = D_k \times D_k \times D_F \times D_F \times M \times N$ . However, the total computational cost of DSC ( $C(\text{DSC})$ ) consists of two parts: Depthwise Convolution and Pointwise Convolution.

$$C(\text{DSC}) = C(\text{Depthwise Convolution}) + C(\text{Pointwise Convolution}) \\ = D_k \times D_k \times D_F \times D_F \times M + M \times N \times D_k \times D_k \quad (1)$$

$$\frac{C(\text{DSC})}{C(\text{routine convolution})} = \frac{1}{N} + \frac{1}{D_F^2} \quad (2)$$

It can be observed that the computational efficiency of DSC (Depthwise Separable Convolution) is significantly higher than that of routine convolution. Therefore, it has been extensively used in lightweight models, such as MobileNet (Howard et al., 2017; Liu et al., 2023), EfficientNet (Tan and Le), FPGA (Field-Programmable Gate Array)-based models (Li et al., 2022), etc. At present, DSC has been widely adopted in various deep learning models (DS-CNN, 2022; Asif et al., 2024). For DSC, we choose the ReLU (Rectified Linear Unit) activation (Boob et al., 2022; Nair and Hinton, 2010). In the deep learning scenario, ReLU has the following advantages over the default sigmoid function. (1) ReLU as an activation function can significantly reduce the computational cost compared to sigmoid, which involves many exponential computations. (2) ReLU can fix the vanishing gradient problem during backpropagation. (3) ReLU can easily generate zero activations for neurons, causing sparsity in the network. Such sparsity helps improve the model's generalization power and reduce the overfitting risk.

Another technical consideration for DSC is the weight initialization method. A well-designed weight initialization scheme can greatly enhance the model convergence. According to Glorot (Glorot and Bengio, 2010), an ideal network weight initialization must satisfy two conditions. (1) Activation variance homogeneity, i.e.,  $\forall i \neq j : \text{Var}(z^i) = \text{Var}(z^j)$ , where  $z^i$  and  $z^j$  are the activation values of the  $i$ -th and  $j$ -th layers of the network; (2) Gradient variance homogeneity, i.e.,  $\forall i \neq j : \text{Var}(\frac{\partial L}{\partial z^i}) = \text{Var}(\frac{\partial L}{\partial z^j})$ , where  $L$  stands for the loss function. According to the above conditions, the authors proposed a uniform initializer, which draws values from  $U(-\sqrt{\frac{6}{fan_{in}+fan_{out}}}, \sqrt{\frac{6}{fan_{in}+fan_{out}}})$ .  $fan_{in}$  and  $fan_{out}$  are the incoming and outgoing numbers of the current neural layer. One

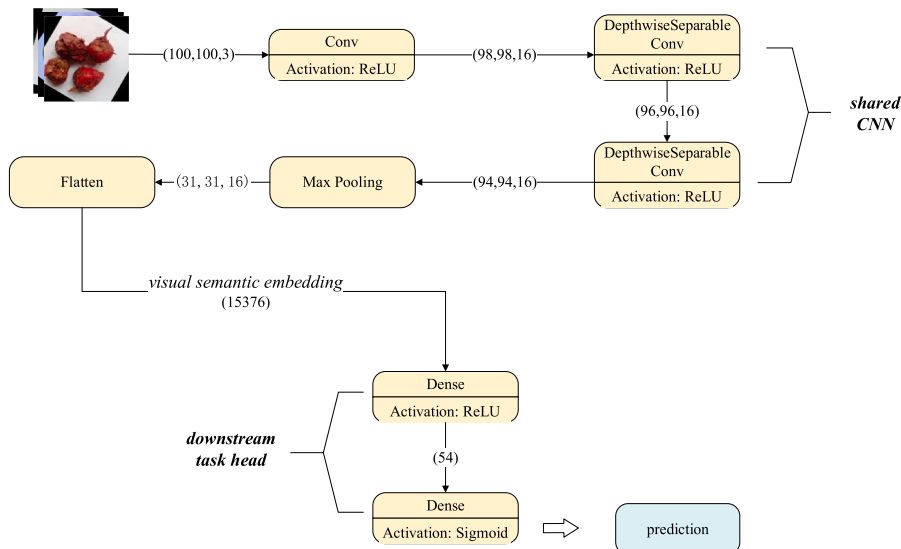


Fig. 1. A base neural network for image classification tasks.

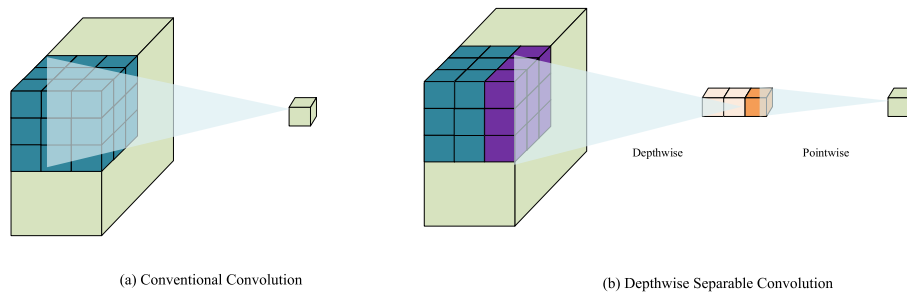


Fig. 2. (a) Conventional convolution. (b) Depthwise separable convolution.

limitation of this initializer is that it requires the activation function to be symmetric around zero, which is not satisfied by the commonly used ReLU. To address this issue, He et al. proposed a new initializer suitable for ReLU (He et al., 2015), a. k.a., He-uniform initialization. It draws values from  $U\left(-\sqrt{\frac{6}{fan_{in}}}, \sqrt{\frac{6}{fan_{in}}}\right)$ , which makes the weight variance only proportional to the incoming number. Experiments have proved this initializer results in better model convergence for ReLU activation. Therefore, we choose He-uniform initialization for the backbone model.

2.2. Multi-task learning model

MTL aims to improve the performance of several different but closely related tasks simultaneously. In deep learning applications, MTL usually defines a neural network as the shared backbone appended by several task heads. A typical MTL example is to use a human face image to predict gender, age, and race simultaneously. In this example, a shared CNN backbone is responsible for extracting common visual features about the input human face. The first task head will be a binary classifier that predicts gender by the visual features. The second task head will be a regressor that returns the predicted age. The third task head will be a multi-classification model that predicts the human race. Such a model design can make use of the mutual knowledge shared by the multiple tasks and usually outperforms single-task learning (STL) models.

To implement MTL for the fruit freshness detection task (denoted as  $T_1$ ), we have to find another related task  $T_2$ . In this study, we choose  $T_2$  as the fruit type classification task. We assume  $T_1$  and  $T_2$  rely on

common visual features of the input fruit image. Based on this assumption, an MTL model is proposed (Fig. 3). In the case study section, we will perform a detailed task-correlation analysis to verify this assumption.

The MTL model contains three components. (1) A shared CNN that returns a semantic embedding of common visual features. (2) A binary classifier task head  $T_1$  for fruit freshness. (3) A parallel multi-classifier task head  $T_2$  for fruit type.

To evaluate the proposed MTL model, we also designed two separate STL models for  $T_1$  and  $T_2$ . The STL model is simply the shared CNN connected with only one task branch. In the next section, we will conduct a case study on an open dataset. Both MTL and STL models will be trained on the same dataset. The test set accuracies of  $T_1$  and  $T_2$  for both scenarios will be collected and compared. Finally, the extracted semantic embeddings generated by the two STL models will be analyzed to evaluate the task correlation.

3. Case study

3.1. Dataset

This study uses an open-source dataset of fresh and rotten fruits (Sultana et al., 2022). The dataset contains 16 classes, i.e., fresh apple, rotten apple, fresh banana, rotten banana, fresh grape, rotten grape, fresh guava, rotten guava, fresh jujube, rotten jujube, fresh pomegranate, rotten pomegranate, fresh strawberry, rotten strawberry, fresh orange, and rotten orange. The dataset can serve two classification tasks.

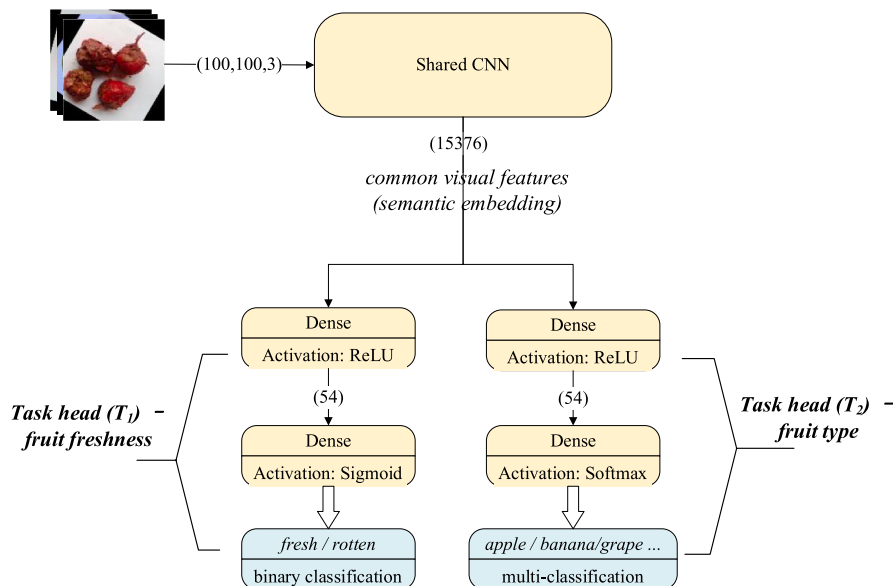


















Fig. 3. A multi-task learning neural network. The model contains a shared CNN module and two task heads. The shared CNN returns a semantic embedding of common visual features. The two task heads are a binary classifier for fruit freshness and a multi-classifier for fruit type.

**Table 1**  
Datasets summary.

Class	Image instances	T <sub>1</sub> label (1 for fresh, 0 for rotten)	T <sub>2</sub> label (1–8 for the eight fruit types)	Example
Fresh apple	734	Y = 1	Y = 1	
Rotten apple	738	Y = 0	Y = 1	
Fresh banana	740	Y = 1	Y = 2	
Rotten banana	736	Y = 0	Y = 2	
Fresh grape	800	Y = 1	Y = 3	
Rotten grape	746	Y = 0	Y = 3	
Fresh guava	797	Y = 1	Y = 4	
Rotten guava	797	Y = 0	Y = 4	
Fresh jujube	793	Y = 1	Y = 5	
Rotten jujube	793	Y = 0	Y = 5	
Fresh orange	796	Y = 1	Y = 6	
Rotten orange	796	Y = 0	Y = 6	
Fresh pomegranate	797	Y = 1	Y = 7	
Rotten pomegranate	798	Y = 0	Y = 7	
Fresh strawberry	737	Y = 1	Y = 8	

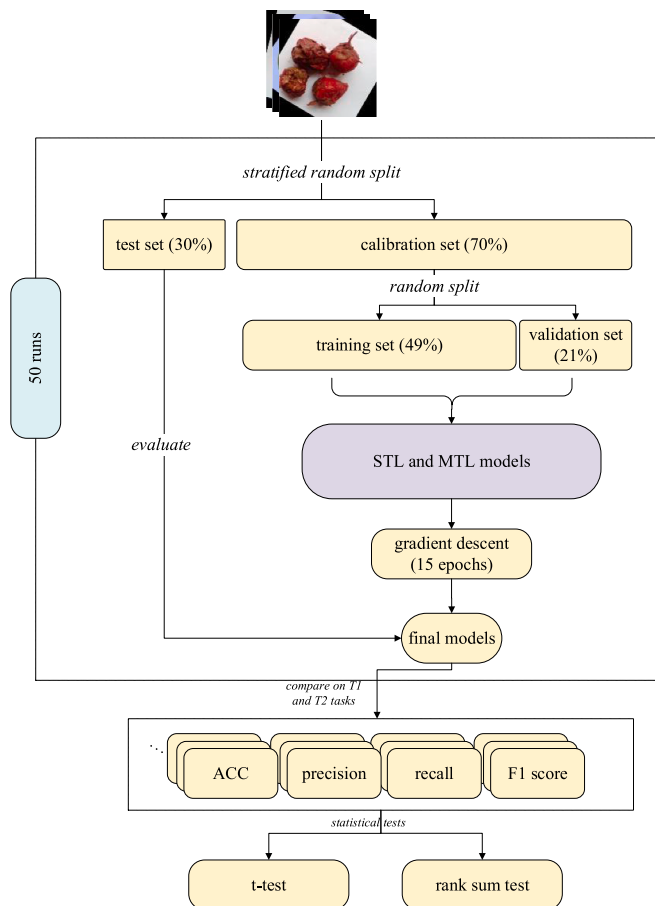
**Table 1 (continued)**

Class	Image instances	T <sub>1</sub> label (1 for fresh, 0 for rotten)	T <sub>2</sub> label (1–8 for the eight fruit types)	Example
Rotten strawberry	737	Y = 0	Y = 8	
<b>Total</b>	<b>12,335</b>			

One is fruit freshness (fresh vs. rotten) and the other is fruit type (apple, banana, grape, etc.). We use them as the T<sub>1</sub> and T<sub>2</sub> task heads in the MTL model. Table 1 is the dataset summary.

3.2. Experiment design

The experiment scheme is shown in Fig. 4. For MTL and STL models, evaluation metrics (classification accuracy, F1 score, etc.) on the test set were collected from 50 runs and statistical tests were performed to get convincing conclusions. In each run, the dataset is randomly split into a 70% calibration set and a 30% test set. During the split, we use the label stratification strategy to ensure that each category is balanced in both the calibration and test sets. The calibration set is further split into a 70% training set and a 30% validation set during model fitting. The calibration set is used to train the deep learning models. The test set is dedicated to model evaluation.



**Fig. 4.** Experimental flowchart.

### 3.3. Loss function and optimizer

In MTL, the total loss is the sum of all sub-tasks. Because  $T_1$  is a binary classification task, we use the binary cross entropy as its loss. For  $T_2$ , as it is a multi-classification task with 8 categories, we use the KLD (Kullback-Leibler Divergence) loss to measure the predicted PMF (Probability Mass Function) with the ground truth. The total loss is defined as follows.

$$L = \text{CrossEntropy}(\hat{y}_{T_1}, y_{T_1}) + \text{KLD}(\hat{p}_{T_2} \| p_{T_2}) = -y_{T_1} \log(1 - \hat{y}_{T_1}) - (1 - y_{T_1}) \log(\hat{y}_{T_1}) + \sum_{y_{T_2} \in L_{T_2}} \left[ \hat{p}_{T_2}(y_{T_2}) \log \left( \frac{\hat{p}_{T_2}(y_{T_2})}{p_{T_2}(y_{T_2})} \right) \right] \quad (3)$$

$$y_{T_1} \in \{0, 1\}; L_{T_2} = \{1, 2, 3, 4, 5, 6, 7, 8\}, p_{T_2} \in \{0, 1\}$$

For a given sample  $x$ ,  $y_{T_1}$  is the ground truth label, and  $\hat{y}_{T_1}$  is the predicted probability of  $P(\hat{y}_{T_1} = y_{T_2} | x)$ .  $L_{T_2}$  is the labels of the multi-classification task  $T_2$ . The  $p_{T_2}$  values for all the  $L_{T_2}$  labels form an OHE (one-hot encoding) encoding vector, which is the ground truth distribution.  $\hat{p}_{T_2}$  is the predicted probability for each label. The  $\hat{p}_{T_2}$  values for all the  $L_{T_2}$  labels form the predicted SoftMax distribution. The KLD part measures the difference between the above two distributions.

To minimize the above loss function, we use the Adam (Adaptive Moment Estimation) (Kingma and Ba, 2017) algorithm. The Adam optimizer uses both momentum and adaptive learning rate, which combines the benefits of RMSProp (Root Mean Squared Propagation) and SGD (Stochastic Gradient Descent) with momentum. Adam can handle non-stationary loss functions with noisy and sparse gradients.

The weight update logic is as follows.

$$g_t = \nabla L(\theta_t) \quad (4)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (5)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (6)$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (7)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (8)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (9)$$

where  $\eta$  is the initial learning rate. In this study, we use  $\eta = 0.001$ .  $\epsilon = 10^{-10}$  is used to prevent the divided-by-zero error.  $\beta_1 = 0.9$  and  $\beta_2 =$

0.999 are forgetting parameters.

### 3.4. Model training

With the above-defined loss function and the Adam optimizer, we conducted the model training on the following hardware and software platforms.

Hardware: Asus ROG GX701.32 GB RAM. Nvidia GeForce RTX 2080 Max-Q with 8 GB dedicated VRAM.

Software: Keras 3.0 deep learning framework with PyTorch backend.

The model is trained with 15 epochs. We use the Adam optimizer and set the learning rate to 0.001. The training curves are shown in Fig. 5. Among all the 15 epochs, we choose the checkpoint with the best validation loss as our final model.

According to the training curve, the model performance gradually improves as the number of epochs. When the number of epochs reaches 15, both  $T_1$  and  $T_2$  have achieved 90% or above accuracy on both the training and validation sets.

### 3.5. Results

The model's performance on the test set is used to evaluate its generalization power. To demonstrate the effectiveness of the proposed MTL model, we also trained its STL counterparts for comparison. For each run, the STL models are trained with the same dataset splitting and optimization settings. Table 2 compares the final test results of STL and MTL. For  $T_1$  (freshness classification), MTL achieves 93.24% accuracy compared to STL's 92.50%. For  $T_2$ , MTL achieves 88.66% accuracy

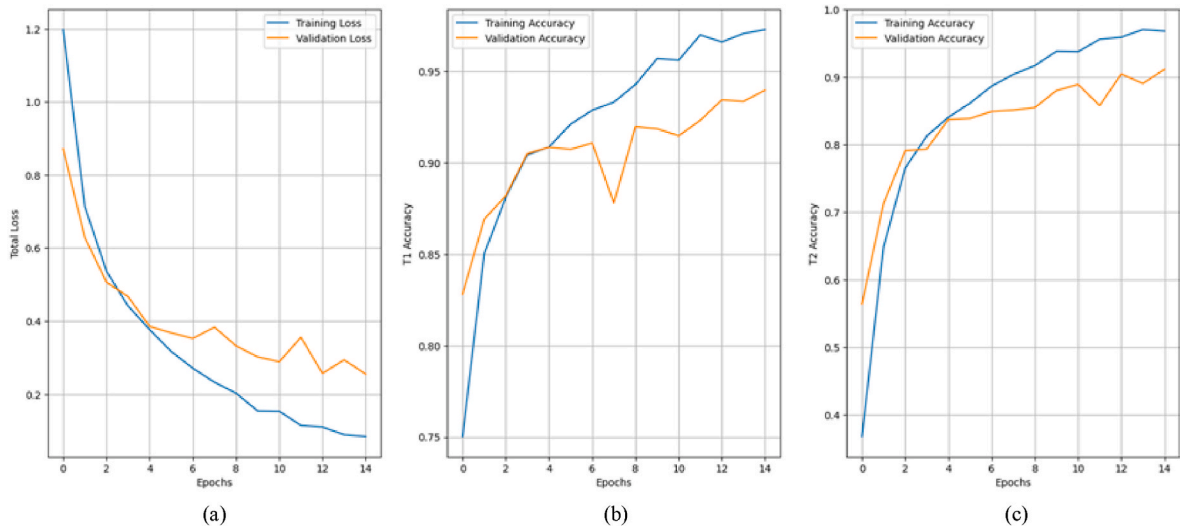


Fig. 5. The training curves. (a) Total loss. (b)  $T_1$  accuracy. (c)  $T_2$  accuracy.



**Table 2**

Comparison of STL and MTL on  $T_1$  and  $T_2$  tasks. The values are measured on the test set and are averaged from 50 runs.

	STL				MTL			
	$\overline{ACC}$	$\overline{F1}$	$\overline{precision}$	$\overline{recall}$	$\overline{ACC}$	$\overline{F1}$	$\overline{precision}$	$\overline{recall}$
$T_1$ task	0.9250	0.9238	0.9344	0.9140	0.9324	0.9315	0.9402	0.9237
$T_2$ task	0.8722	0.8714	0.8743	0.8723	0.8866	0.8860	0.8881	0.8867

**Table 3**

The results of the hypothesis tests of STL and MTL on  $T_1$  and  $T_2$  tasks.

	$t$ -test		rank sum test	
	statistic	p-value	statistic	p-value
$T_1$ task	-4.33	$7.4 \times 10^{-5}$	-4.33	$1.5 \times 10^{-5}$
$T_2$ task	-4.48	$4.5 \times 10^{-5}$	-5.08	$3.8 \times 10^{-7}$

compared to STL's 87.22%, and MTL also demonstrates advantages in terms of F1 score, precision, and recall (see Table 2).

Furthermore, we performed parametric  $t$ -tests and non-parametric hypothesis tests on the collected metrics to verify if the aforementioned findings were statistically significant. The  $t$ -test and rank sum test report that there is a significant difference between the test accuracies of STL and MTL, indicating that MTL outperforms STL (see Table 3).

### 3.6. Misclassification analysis

This section will display some typical instances of mispredicted samples and explore the reasons why the MTL model fails. Fig. 6 shows three typical misclassification cases. In Fig. 6(a), a rotten guava is misclassified as an orange. This is mainly due to the guava's poor freshness, as its color does not exhibit the normal greenish hue and is mistaken as an orange. Similarly, in Fig. 6(b), a rotten guava is misclassified as a banana. Similar to the first case, the rotten guava has a yellow color and its surface resembles the skin or cross-section of a stale banana, leading to misclassification. In Fig. 6(c), an apple is misclassified as a pomegranate. This is mainly due to the high similarity in appearance between "a pile of apples" and a pomegranate, as they share similar visual features in color and shape.

We summarize the main reasons for misclassifications as follows: (1) When the freshness of some fruits is low, there is a significant color change, making them more similar to other types of fruits; (2) Some fruits naturally have similar features, such as "a pile of apples" and a pomegranate; (3) The low image resolution worsens the misclassification as visual details are insufficient.

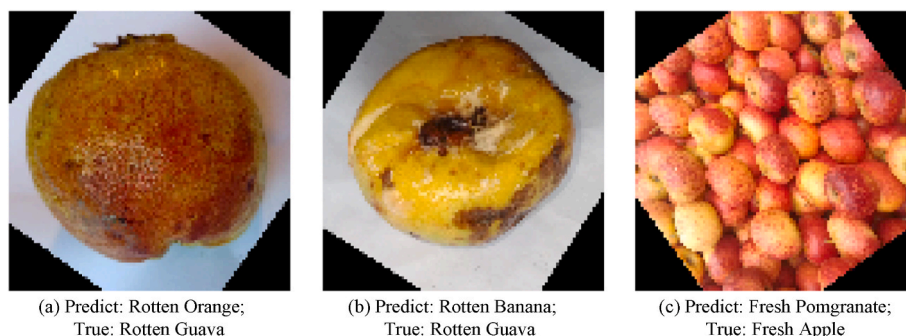
Therefore, in future work, we will improve the model's performance through various means: (1) Use higher resolution images and more complex models; (2) For fruits with similar appearance features, collect more samples and add more task heads.

### 3.7. Task correlation analysis

In the above case study, we have shown MTL's effectiveness by comparing it with the STL counterparts. This section will further investigate why MTL works for the target dataset. The success of MTL depends on the correlation or similarity between the involved sub-tasks. In the context of STL, the output vectors from the shared CNN of different task models should exhibit a high degree of similarity. This is often indicative of a strong correlation between the tasks, thereby justifying the sharing of the network structure and the MTL approach. To measure such a relationship, we define the following task-correlation measurement workflow (Fig. 7). The workflow includes three steps. (1) First, we take the intermediate vectors produced by the shared CNN of STL models. These vectors can be seen as an equal-length embedding containing extracted visual information by each STL model. In this study, the embedding is a 15,376-long vector returned from the last "flatten" layer in the shared CNN. For each STL model, we sort the features in descending order by their average value on the entire dataset. In this way, we assume the embeddings from the two tasks are aligned in the same order where significant visual features come before trivial ones. (2) The second step involves calculating a task-correlation metric. This metric can be cosine similarity or Pearson's correlation coefficient. Cosine similarity has been extensively used to measure the semantic similarity between word embeddings in natural language processing (NLP). The cosine similarity is proved to be an unbiased and consistent estimator and is insensitive to sample size (Chou and Hsu, 2018). It is defined as follows.

$$\text{CosSim}(x, y) = \frac{x \bullet y}{\|x\| \bullet \|y\|} \quad (10)$$

Pearson's correlation coefficient is the more traditional way to measure correlations. It is defined as follows.



**Fig. 6.** Misclassification examples.

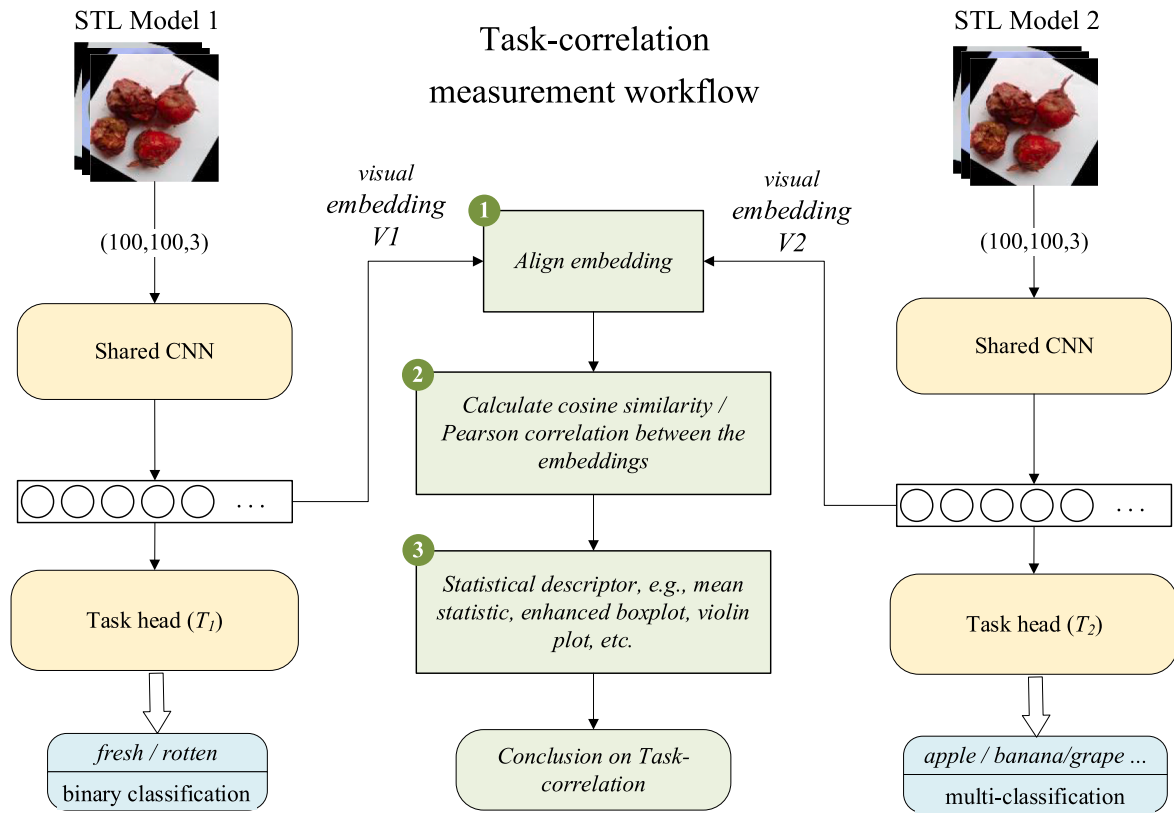


Fig. 7. Use STL models to measure task correlation. The workflow contains three steps. (1) Align the visual embeddings extracted by the two STL models. (2) Calculate per-sample task-correlation metrics, e.g., cosine similarity and Pearson correlation between the two embedding vectors. (3) Use various statistical descriptors to evaluate the correlation metric of all samples in the dataset.

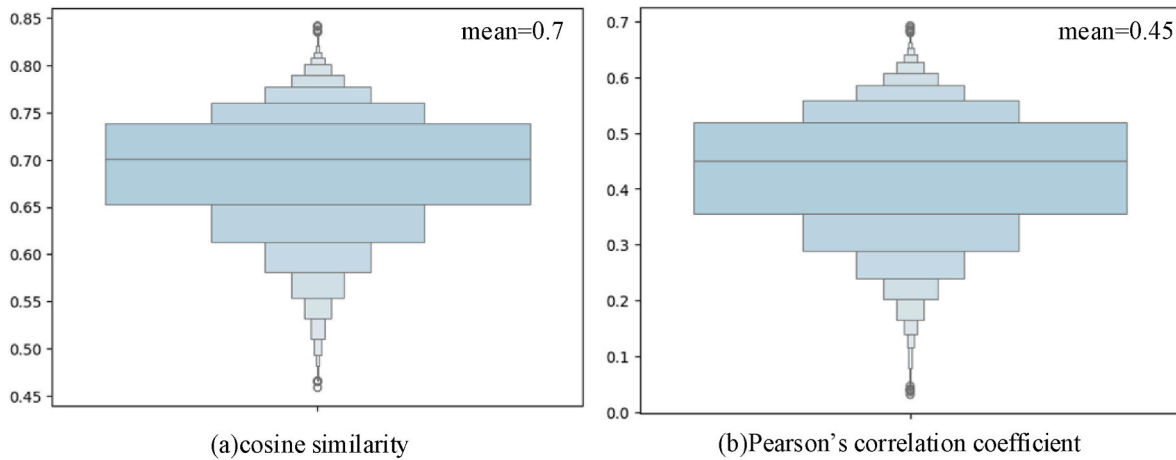


Fig. 8. The enhanced boxplots of cosine similarity and Pearson's correlation coefficient on the entire dataset.

$$Corr(x, y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}} = \frac{\langle x - \bar{x}, y - \bar{y} \rangle}{\|x - \bar{x}\| \cdot \|y - \bar{y}\|} = CosSim(x - \bar{x}, y - \bar{y}) \tag{11}$$

From the above equations, we can see that Pearson's correlation coefficient is a "demeaned" version of cosine similarity. In this study, both metrics will be used. (3) Finally, after calculating the correlation metrics on all the samples, we can use various statistical descriptors to evaluate the metrics on the entire dataset, such as the mean statistic, the frequency histogram, the enhanced boxplot, or the violin plot.

Fig. 8 shows the overall distribution of cosine similarity and Pearson's correlation coefficient on the entire dataset. The two vectors exhibit a high level of similarity under both metrics, with a mean value of 0.7 and 0.45.

## 4. Discussion

### 4.1. Application prospects

This study proposes an MTL technical framework for the fruit freshness related tasks. This study reveals that for the task of fruit freshness detection, sharing a portion of the network weights has improved the detection efficiency compared to single-task learning. This technique may find its use in more agricultural applications. For example, multi-task learning can be used to achieve precise identification and harvesting of high-quality fruits in orchards where mixed fruit types are involved. It can also be utilized to identify crops and weeds in fields for precise harvesting and weed control.

### 4.2. Network architecture optimization

In this study, one interesting point worth further investigation is how to find the best neural network architecture for the target tasks. Unlike the general NAS (Neural Architecture Search) problem, MTL has extra concerns, e.g., how many layers should be assigned to the shared backbone network and where should the task heads branch off. For complex MTL tasks, it might be worth to try branch off at different points for different task heads, depending on each task's requirement on features' semantic granularity.

### 4.3. Balance between subtasks

Another core concern for MTL is how to balance the multiple subtasks. In this study, the MTL loss function is simply the sum of the two sub-task losses, i.e., we assign equal importance. From the actual experiment, this equal-importance assumption works and we have achieved the desired result. However, for certain problems, we may need to assign different weights. For example, if one task loss is significantly larger than others, it will dominate the training. In this case, we will have to rescale all sub-losses to a comparable level. Even after such rescaling, during the training, the sub-losses may still have different convergence speeds. This requires sophisticated training hyperparameters to make all sub-tasks have balanced gradients and converge at a synchronized speed.

### 4.4. Other task-correlation metrics

In this study, we use cosine similarity and Pearson's correlation coefficient between different tasks' intermediate visual embeddings to measure their correlation. Besides these numeric metrics, we may also consider visual probe techniques in the XAI (Explainable Artificial Intelligence) domain, e.g., use grad-CAM (Gradient-weighted Class Activation Mapping) to generate heatmaps of extracted features. These techniques may provide a more intuitive explanation of the task correlation.

### 4.5. Future research

Besides the above concerns, we will focus on the following aspects:

- (1) We will expand the model to include more fruit or other food datasets, such as FruitQ (Abayomi-Alli et al., 2024), and optimize the model based on real-world conditions;
- (2) We will attempt to explore and experiment with different network architectures as shared CNN and study the impact of these different network architectures on the model;
- (3) We will explore other multi-task learning applications in the agricultural domain, such as precision harvesting of crops;
- (4) Develop a GUI frontend for the model to facilitate users in analyzing their datasets.

## 5. Conclusion

This study proposed a multi-task learning (MTL) model for CV (computer vision)-based fruit freshness detection. A case study on an open dataset has demonstrated MTL outperforms its STL counterpart. We also measured the task correlation using the intermediate visual embeddings. We have proven MTL can effectively share related tasks' underlying feature extraction process and achieve optimized coordinated training. The proposed MTL model can be used for fruit freshness detection in various automatic harvesting and supply chain monitoring applications.

### Data and software availability

The data is a published open image set (<https://doi.org/10.1016/j.dib.2022.108552>).

The code has been published to the CodeOcean (<https://doi.org/10.24433/CO.4402502.v1>).

### CRediT authorship contribution statement

**Yinsheng Zhang:** Conceptualization, Writing – review & editing, Data curation, Funding acquisition. **Xudong Yang:** Software, Methodology, Writing – original draft, Visualization. **Yongbo Cheng:** Project administration. **Xiaojun Wu:** Methodology, Data curation. **Xiulan Sun:** Supervision, Data curation. **Ruiqi Hou:** Visualization, Data curation. **Haiyan Wang:** Supervision, Project administration, Funding acquisition.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgments

This work was supported by the National Natural Science Foundation of China (62376249, 91746202) and the Ministry of Science and Technology of the People's Republic of China (2023YFD1000400).

### References

- Abayomi-Alli, O.O., Damaševičius, R., Misra, S., Abayomi-Alli, A., 2024. FruitQ: a new dataset of multiple fruit images for freshness evaluation. *Multimed. Tool. Appl.* 83 (4), 11433–11460. <https://doi.org/10.1007/s11042-023-16058-6>.
- Amrani, A., Diepeveen, D., Murray, D., Jones, M.G.K., Soheli, F., 2024. Multi-task learning model for agricultural pest detection from Crop-plant imagery: a bayesian approach. *Comput. Electron. Agric.* 218, 108719 <https://doi.org/10.1016/j.compag.2024.108719>.
- Asif, S., Zhao, M., Tang, F., Zhu, Y., 2024. DCDS-net: deep transfer network based on depth-wise separable convolution with residual connection for diagnosing gastrointestinal diseases. *Biomed. Signal Process Control* 90, 105866. <https://doi.org/10.1016/j.bspc.2023.105866>.
- Baranowski, P., Mazurek, W., Pastuszka-Woźniak, J., 2013. Supervised classification of bruised apples with respect to the time after bruising on the basis of hyperspectral



- imaging data. *Postharvest Biol. Technol.* 86, 249–258. <https://doi.org/10.1016/j.postharvbio.2013.07.005>.
- Birwal, P., S, P., Bk, Y., 2015. Importance of objective and subjective measurement of food quality and their inter-relationship. *J. Food Process. Technol.* 6 (9) <https://doi.org/10.4172/2157-7110.1000488>.
- Boob, D., Dey, S.S., Lan, G., 2022. Complexity of training ReLU neural network. *Discrete Optim.* 44, 100620 <https://doi.org/10.1016/j.disopt.2020.100620>.
- Chollet, F. Xception, 2017. Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, pp. 1800–1807. <https://doi.org/10.1109/CVPR.2017.195>.
- Chou, E.P., Hsu, S.-M., 2018. Cosine similarity as a sample size-free measure to quantify phase clustering within a single neurophysiological signal. *J. Neurosci. Methods* 295, 111–120. <https://doi.org/10.1016/j.jneumeth.2017.12.007>.
- DS-CNN, 2022. A pre-trained xception model based on depth-wise separable convolutional neural network for finger vein recognition. *Expert Syst. Appl.* 191, 116288 <https://doi.org/10.1016/j.eswa.2021.116288>.
- Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. *PMLR* 9, 249–256.
- Gong, S., Zhang, J., Zheng, X., Li, G., Xing, C., Li, P., Yuan, J., 2023. Recent design strategies and applications of organic fluorescent probes for food freshness detection. *Food Res. Int.* 174, 113641 <https://doi.org/10.1016/j.foodres.2023.113641>.
- He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV). IEEE, Santiago, Chile, pp. 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>.
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv*. <https://doi.org/10.48550/arXiv.1704.04861>.
- Huang, Z., Wu, J., Xie, F., 2021. Automatic surface defect segmentation for hot-rolled steel strip using depth-wise separable U-shape network. *Mater. Lett.* 301, 130271 <https://doi.org/10.1016/j.matlet.2021.130271>.
- Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization.
- Li, G., Zhang, J., Zhang, M., Wu, R., Cao, X., Liu, W., 2022. Efficient depthwise separable convolution accelerator for classification and UAV object detection. *Neurocomputing* 490, 1–16. <https://doi.org/10.1016/j.neucom.2022.02.071>.
- Liu, Y., Wang, Z., Wang, R., Chen, J., Gao, H., 2023. Flooding-based MobileNet to identify cucumber diseases from leaf images in natural scenes. *Comput. Electron. Agric.* 213, 108166 <https://doi.org/10.1016/j.compag.2023.108166>.
- Lu, H., Xu, S., Wang, J., 2023. Multi-dataset fusion for multi-task learning on face attribute recognition. *Pattern Recogn. Lett.* 173, 72–78. <https://doi.org/10.1016/j.patrec.2023.07.015>.
- Nair, V., Hinton, G.E., 2010. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, Haifa, Israel, pp. 807–814.
- Salim, E., 2023. Suhajito. Hyperparameter optimization of YOLOv4 tiny for palm oil fresh fruit bunches maturity detection using genetics algorithms. *Smart Agric. Technol.*, 100364 <https://doi.org/10.1016/j.atech.2023.100364>.
- Sultana, N., Jahan, M., Uddin, M.S., 2022. An extensive dataset for successful recognition of fresh and rotten fruits. *Data Brief* 44, 108552. <https://doi.org/10.1016/j.dib.2022.108552>.
- Sun, L., Zhang, Y., Liu, T., Ge, H., Tian, J., Qi, X., Sun, J., Zhao, Y., 2023. A collaborative multi-task learning method for BI-RADS category 4 breast lesion segmentation and classification of MRI images. *Comput. Methods Progr. Biomed.* 240, 107705 <https://doi.org/10.1016/j.cmpb.2023.107705>.
- Tan, M.; Le, Q. V. EfficientNet: rethinking model scaling for convolutional neural networks. *arXiv September 11, 2020*. <http://arxiv.org/abs/1905.11946> (accessed 2023-November-15).
- Tang, S., Yu, X., Cheang, C.F., Liang, Y., Zhao, P., Yu, H.H., Choi, I.C., 2023. Transformer-based multi-task learning for classification and segmentation of gastrointestinal tract endoscopic images. *Comput. Biol. Med.* 157, 106723 <https://doi.org/10.1016/j.combiomed.2023.106723>.
- Ventura-Aguilar, R.I., Bautista-Baños, S., Hernández-López, M., Llamas-Lara, A., 2021. Detection of alternaria alternata in tomato juice and fresh fruit by the production of its biomass, respiration, and volatile compounds. *Int. J. Food Microbiol.* 342, 109092 <https://doi.org/10.1016/j.ijfoodmicro.2021.109092>.
- Yuan, Y., Chen, X., 2024. Vegetable and fruit freshness detection based on deep features and principal component analysis. *Curr. Res. Food Sci.* 8, 100656 <https://doi.org/10.1016/j.crfs.2023.100656>.
- Zhao, Y., Wang, X., Che, T., Bao, G., Li, S., 2023. Multi-task deep learning for medical image computing and analysis: a review. *Comput. Biol. Med.* 153, 106496 <https://doi.org/10.1016/j.combiomed.2022.106496>.
- Zhong, X., Zhang, M., Tang, T., Adhikari, B., Ma, Y., 2023. Advances in intelligent detection, monitoring, and control for preserving the quality of fresh fruits and vegetables in the supply chain. *Food Biosci.* 56, 103350 <https://doi.org/10.1016/j.fbio.2023.103350>.
- Zhou, Y., Chen, H., Li, Y., Liu, Q., Xu, X., Wang, S., Yap, P.-T., Shen, D., 2021. Multi-task learning for segmentation and classification of tumors in 3D automated breast ultrasound images. *Med. Image Anal.* 70, 101918 <https://doi.org/10.1016/j.media.2020.101918>.