ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

TWO-STAGE HYBRID YOLO-EFFICIENTNET FRAMEWORK FOR APPLE DISEASE DETECTION

Srinivas Dussa[1], Praveen Kumar Saddala[2]
1 PG Scholar, Department of Electronics and Communication Engineering, JNTUH University College of Engineering Jagtial, Jagtial, Telangana. 505501
2 Associate Professor, Department of Electronics and Communication Engineering, JNTUH University College of Engineering Jagtial,
Jagtial, Telangana. 505501
dussasrinivas028@gmail.com

***Abstract*— Accurate recognition of apple fruit diseases from orchard images is difficult because fruits may be partly occluded and disease symptoms can be visually similar. This paper presents a two-stage hybrid vision pipeline that separates fruit localization from disease classification. The first stage uses a fine-tuned YOLO11n detector, optimized through a 50-iteration genetic hyperparameter search, to localize apple regions in full-scene images. Detected regions are cropped, resized to 224 x 224 pixels and passed to a pre-trained EfficientNet-B0 classifier. To reduce class-frequency bias, the classifier backbone is unfrozen and fine-tuned for ten epochs using class-weighted cross-entropy. The dataset combines 4,154 images from two Roboflow repositories and represents eight classes: Anthracnose, Black Pox, Black Rot, Blotch, Healthy, Powdery Mildew, Rot and Scab. Fine-tuned YOLO11n achieved 94.22% precision, 92.60% recall, 96.86% mAP@0.50 and 78.63% mAP@0.50:0.95, with detector latency of 3.0 ms per image on an NVIDIA GeForce RTX 2060. Fine-tuned EfficientNet-B0 reached 96.49% peak validation accuracy. The findings indicate that separating localization and classification is a promising approach; independent field testing and end-to-end latency measurement are still required.**

Keywords*— Apple fruit disease; object detection; YOLO11n; EfficientNet-B0; transfer learning; precision agriculture.

I. INTRODUCTION

Apple (Malus domestica), as a major fruit crop, is of interest and the early identification of fruit diseases has implications for market quality and losses that can be avoided. Disease assessment is in practice usually done by visual inspection. This is a time-consuming method if there are a large number of fruits to be checked and it can be subjective depending on the experience of the observer. Automated image analysis can be used to help with this process; however, there are challenges that often arise with images taken in the field that are not necessarily present in the laboratory. Fruits can be partially obscured by foliage or branches, light may vary from one spot to another within the scene and various disorders can cause similar spots or discoloration and surface patterns.

It has been proved in previous research that deep learning can be used to aid in the diagnosis of apple diseases. [1] proposed a multi-scale dense classification network for typical apple diseases with 94.31% and 94.74% classification accuracy for two suggested models. In their study [2], they analyzed the image-processing and artificial-neural-network techniques used to classify four classes of apples, such as scab, bitter rot, black rot and healthy fruit. More recent work has considered detection under complex field conditions. A YOLOv11-based detector for apple leaf diseases was developed by [3] and the detection and segmentation of apple scab on fruit was done using YOLO11 for preharvest and postharvest in [4]. The utility of both image classification and object detection is shown from these studies and the two tasks assume different things. Typically a classifier works best if the target fruit has already been separated from the background. An Object Detector must also be able to detect the different types of disease while detecting the object, in addition to localizing the object in the whole image. This paper fills this gap in practice by performing fine-grained disease recognition on the obtained regions by using a separate classifier and the obtained region is identified using a detector. The proposed system has three main contributions. First, it is a multi-class dataset of 4,154 full scene images and 4,419 crops identified in the fruit region across eight health and disease classes that come from two public Roboflow image collections. Second, it compares baseline YOLO11n and YOLO11s detectors, and optimizes the hyperparameters of YOLO11n using the genetic algorithm. Third, it compares five classification architectures and fine-tunes the EfficientNet-B0 model with class-weighted cross-entropy. A prototype using flask is also included to enable users to upload images or use the camera to feed images into the system and display the diagnostic results.

II. RELATED WORK

A. Deep Learning for Apple Disease Classification

Deep convolutional networks learn image features from pixels and can model disease patterns that are difficult to represent through manually designed colour or texture

descriptors. [1] combined DenseNet features with multi-scale connections to diagnose eleven typical apple disease image categories. Their results support the use of feature reuse and multi-scale representation for visually similar symptoms. [2] investigated an image-processing pipeline with an artificial neural network for four apple fruit classes and highlighted the value of image-based diagnosis outside purely leaf-focused classification tasks.

The present study uses EfficientNet-B0 as its primary classifier. EfficientNet was designed around compound scaling of network depth, width and input resolution, providing a structured way to balance model capacity and computational cost [5]. ResNet and DenseNet are included as baselines because residual connections and dense feature reuse have been widely used to train deeper image-recognition networks [6], [7].

B. Object Detection in Field Images

YOLO models make predictions about object location and class scores in a single forward pass and are often utilized when a fast localization is needed. YOLO11 is a model family released in September 2024, and according to Ultralytics, does not have a separate formal research paper, so software documentation is used for the implementation reference (Ultralytics, 2024). For apple disease detection in complex field settings, [3] re-imagined YOLOv11 for leaf disease detection. In the fruit detection and scab-lesion segmentation, [4] used the YOLO11 process. These studies emphasize the need to test in variable field backgrounds and to break down object localization and detailed symptom analysis tasks when both are required.

C. Research Gap

One model might need to be used simultaneously to solve two different problems: the first is locating a fruit in an uncropped orchard image, and the second is distinguishing fine disease symptoms on the fruit. The image level classification techniques alone cannot address the localization problem and a detector trained to recognize multiple disease classes may not be able to make fine-grained discrimination on the symptoms. The framework analyzed in this paper separates these tasks and measures each one according to a metric that fits the task. The goal is not to propose the new detector or new architecture of the classifier itself, but to use a task-specific combination of YOLO11n and EfficientNet-B0 on an apple-fruit dataset with 8 classes.

III. MATERIALS AND METHODS

A. Dataset Collection and Curation

The dataset has been collected in a combination of two public apple-image repositories on Roboflow Universe [9]. The first source had annotations indicating instance-segmentation, and the second had boxes: bounding boxes of objects. Before combining the two sources, 145 polygon annotations were converted to bounding boxes. The final collection consisted of 4,154 images of whole scenes and 4,420 images of fruit instances. The bounding-box regions were cropped and hence 4,419 region-of-interest (ROI) images were extracted for classification.

The classification task has eight categories: Anthracnose, Black Pox, Black Rot, Blotch, Healthy, Powdery Mildew, Rot and Scab. The class distribution of the 4,419 cropped regions is presented in Table 1. Blotch and Rot make up a relatively high percentage of the crops while there are fewer examples of Anthracnose, Black Pox, Black Rot and Powdery Mildew. This imbalance provided motivation for employing class-weighted loss for fine-tuning classifiers.

Table. 1. Distribution of the cropped apple-fruit dataset

Class	ROI crops
Anthracnose	219
Black Pox	239
Black Rot	232
Blotch	1,127
Healthy	771
Powdery Mildew	239
Rot	954
Scab	638
Total	4,419

Note: The class totals sum to 4,419 cropped fruit regions, as reported in the project thesis.

B. Data Partitioning and Preprocessing

For Stage 1, the 4,154 full-scene images were divided into 3,323 training images and 831 validation images, corresponding to an 80:20 split. The classifier dataset was also described as using an 80:20 training-validation split. The confusion matrix reported for the fine-tuned EfficientNet-B0 contains 884 validation crops, which is consistent with 20% of the 4,419 ROI images. Accordingly, the classifier results in this paper are treated as validation results rather than results from an independently held-out external test set.

For object detection, input images were resized to 640 x 640 pixels. Training augmentation included mosaic augmentation, mix-up, random scaling, translation, horizontal flipping and selected Albumentations transformations, including CLAHE, blur, median blur and grayscale conversion. Mosaic augmentation was disabled during the final ten training epochs to stabilize bounding-box regression. For classification, each cropped fruit region was resized to 224 x 224 pixels. Horizontal flipping, rotations within +/-15 degrees and colour jittering were applied. Input channels were normalized using the ImageNet mean [0.485, 0.456, 0.406] and standard deviation [0.229, 0.224, 0.225].

C. Proposed Two-Stage Framework

The proposed pipeline separates spatial localization and fine-grained disease classification. In Stage 1, the full input image is passed to a fine-tuned YOLO11n detector. The detector predicts bounding boxes for apple fruits in the scene. Each detected region is then cropped from the source image and passed through the classification preprocessing pipeline. In Stage 2, EfficientNet-B0 assigns the cropped region to one of the eight health or disease categories. The predicted class and confidence value are returned to the application interface. The overall processing sequence is illustrated in Fig. 1.

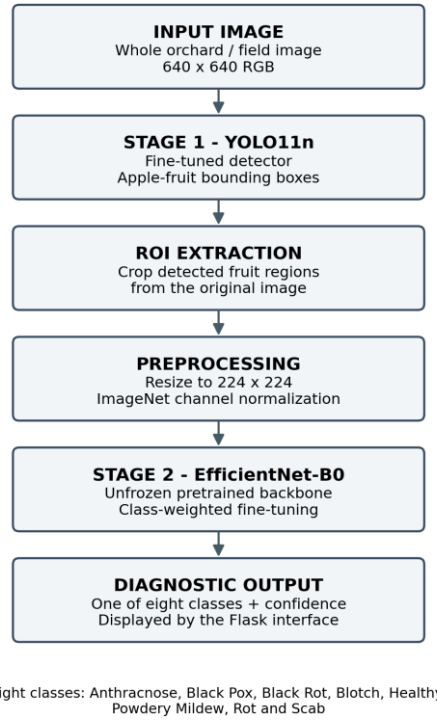


Fig. 1. Two-stage hybrid pipeline for apple-fruit localization and disease classification.

This separation allows the detector to concentrate on locating fruit regions in the presence of background clutter, while the classifier focuses on the visual patterns within each cropped fruit. The bounding-box crop is taken from the original image and resized to the classifier input dimensions before channel normalization. The two stages are trained and evaluated separately, and their reported metrics are therefore not interchangeable.

D. Stage 1: YOLO11n Detection and Optimization

YOLO11n was selected as the lightweight localization model. Baseline YOLO11n and YOLO11s were used for comparison. The YOLO11n training configuration was optimized through a 50-iteration genetic hyperparameter search. The selected detector was trained for 100 epochs at an input resolution of 640 x 640 pixels and a batch size of 16. AdamW was used with a learning rate of 0.000833, momentum of 0.9000 and weight decay of 0.00026. Automatic mixed precision was enabled. The implementation used the Ultralytics YOLO engine (version 8.4.94). AdamW follows the decoupled weight-decay formulation described by [10].

E. Stage 2: EfficientNet-B0 Classification

Five candidate classifiers were benchmarked on the cropped ROI images: a Custom CNN, DenseNet-201, ResNet-50, YOLO11 Classifier and EfficientNet-B0. All models used 224 x 224 input images and a batch size of 32 for the comparative evaluation. EfficientNet-B0 achieved the highest baseline validation accuracy and was selected for fine-tuning. The pre-trained backbone was unfrozen and trained for ten additional epochs using the Adam optimizer. Class-weighted cross-entropy loss was applied, with class weights computed from sample frequencies. The random seed was set to 42 for the fine-tuning procedure.

F. Web Application Prototype

A prototype based on the Flask framework was created to showcase the integrated inference workflow. An uploaded image or image captured from the camera input is passed to the two-stage pipeline and the output diagnostic class and confidence is shown in the interface. The project also outlines user authentication and scan-history storage using SQLite. The prototype in the reported experiments is not evaluated using a formal usability study, but is a demonstration of the link between model inference and user interaction.

G. Evaluation Measures

Detection performance was assessed using precision, recall, mean average precision at an intersection-over-union threshold of 0.50 (mAP@0.50), and mean average precision averaged over thresholds from 0.50 to 0.95 (mAP@0.50:0.95). Precision is TP/(TP+FP), and recall is TP/(TP+FN), where TP, FP and FN denote true-positive, false-positive and false-negative detections. Classification performance was assessed using overall accuracy and, for the baseline comparison, weighted precision, weighted recall and weighted F1-score. Accuracy is the number of correctly classified validation samples divided by the total number of validation samples. Detector inference latency is reported in milliseconds per input image and should not be interpreted as the latency of the complete two-stage system.

H. Computing Environment

Experiments were conducted on a workstation with an NVIDIA GeForce RTX 2060 GPU with 6 GB of video memory, running Windows 11 Professional. The reported software environment included Python 3.12.2, PyTorch 2.11.0+cu128, Ultralytics 8.4.94, OpenCV, Torchvision 0.16, Albumentations, NumPy and scikit-learn.

IV. RESULTS AND DISCUSSION

A. Dataset Profile

The curated data contain 4,154 full-scene images, 4,420 annotated detection instances and 4,419 cropped fruit regions. Table 1 shows that the classification dataset is imbalanced. Blotch is the largest class with 1,127 crops, followed by Rot with 954 and Healthy with 771. Anthracnose, Black Pox, Black Rot and Powdery Mildew each contain fewer than 250 crops. Class-weighted cross-entropy was therefore used during the final EfficientNet-B0 fine-tuning stage to reduce the influence of class-frequency differences on the learning objective.

B. Stage 1 Detection Results

Table 2 compares the two baseline detectors with the fine-tuned YOLO11n. Fine-tuned YOLO11n achieved 94.22% precision and 92.60% recall, with mAP@0.50 of 96.86% and mAP@0.50:0.95 of 78.63%. Its mAP@0.50 improved by 3.76 percentage points over baseline YOLO11n and by 3.46 percentage points over YOLO11s. The stricter mAP@0.50:0.95 measure also increased from 69.90% for baseline YOLO11n to 78.63% after fine-tuning.

Table. 2. Stage 1 object-detection benchmark

Detector	P (%)	R (%)	mAP50	mAP50:95	ms/img	Params
YOLO11n (baseline)	92.4	87.3	93.10	69.90	2.5	2,591,400

Detector	P (%)	R (%)	mAP50	mAP50:95	ms/img	Params
YOLO11s	91.3	88.1	93.40	67.60	5.8	~9,400,000
Fine-tuned YOLO11n	94.22	92.60	96.86	78.63	3.0	2,583,712

Note: Latency refers to Stage 1 detector inference on an NVIDIA GeForce RTX 2060 GPU; it is not end-to-end application latency.

The reported detector latency was 3.0 ms per image on the RTX 2060. Baseline YOLO11n was slightly faster at 2.5 ms, while YOLO11s required 5.8 ms. Thus, the optimized detector did not have the lowest latency among the three variants; rather, it improved localization metrics while remaining faster than YOLO11s and retaining a similar parameter count to baseline YOLO11n. The latency figure applies to Stage 1 only because the thesis does not provide a separately measured end-to-end latency for ROI cropping, EfficientNet-B0 inference and application response.

Class-wise detector results are shown in Table 3. Powdery Mildew recorded the highest mAP@0.50 at 99.5%, followed by Black Rot at 99.4% and Scab at 99.2%. Black Pox had the lowest mAP@0.50 at 88.1% and the lowest precision at 79.5%. This pattern is consistent with the visual difficulty of distinguishing small, dark lesions and overlapping symptoms in orchard images. Blotch also showed lower recall than several other categories, indicating that some annotated instances were missed by the detector.

Table. 3. Class-wise detection performance of fine-tuned YOLO11n

Class	Imgs	Inst.	P (%)	R (%)	mAP50	mAP50:95
Anthracnose	200	219	95.8	94.4	98.5	75.5
Black Pox	170	239	79.5	80.8	88.1	59.6
Black Rot	231	232	99.1	97.0	99.4	75.5
Blotch	981	1,128	93.8	83.4	93.5	76.8
Healthy	768	771	94.5	95.5	98.4	89.1
Powdery Mildew	239	239	99.3	99.2	99.5	72.9
Rot	892	954	94.7	94.1	98.3	84.6
Scab	636	638	97.0	96.6	99.2	95.1

C. Stage 2 Classification Benchmark

The baseline classifier comparison is summarized in Table 4. The Custom CNN achieved 75.45% accuracy. DenseNet-201 and ResNet-50 reached 90.72% and 93.67%, respectively, while the YOLO11 Classifier achieved 94.23%. EfficientNet-B0 produced the best baseline result, with 95.14% accuracy, 95.65% weighted precision, 95.14% weighted recall and 95.14% weighted F1-score. These results motivated its selection for further fine-tuning.

Table. 4. Baseline Stage 2 classification benchmark

Classifier	Acc. (%)	W. Prec. (%)	W. Recall (%)	W. F1 (%)
Custom CNN	75.45	75.84	75.45	75.04
DenseNet-201	90.72	91.11	90.72	90.77
ResNet-50	93.67	93.84	93.67	93.65
YOLO11 Classifier	94.23	94.39	94.23	94.22
EfficientNet-B0	95.14	95.65	95.14	95.14

After unfreezing the EfficientNet-B0 backbone and applying class-weighted cross-entropy for ten epochs, peak validation accuracy increased from 95.14% to 96.49%, an absolute improvement of 1.35 percentage points. The reported validation loss at the selected point was 0.2710. The model comparison and fine-tuning result are summarized in Fig. 2.

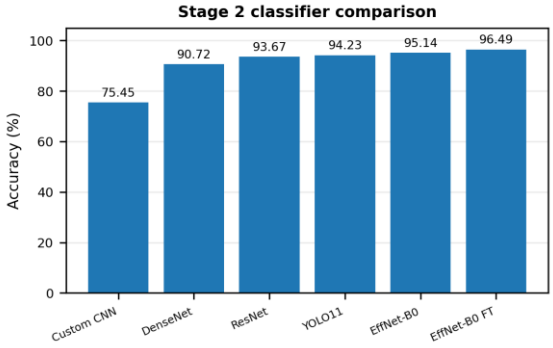


Fig. 2. Baseline classifier accuracy and fine-tuned EfficientNet-B0 validation accuracy.

D. Confusion Matrix Analysis

The fine-tuned classifier confusion matrix is reproduced in Fig. 3. It contains 884 validation samples, of which 853 were correctly classified, giving an accuracy of 96.49%. Powdery Mildew and Rot were classified correctly for all validation samples represented in the matrix. The largest off-diagonal count was the assignment of 11 Blotch samples to Healthy. Black Rot had 41 correct predictions out of 47 samples, giving the lowest class-wise recall in the matrix (87.23%); five Black Rot samples were predicted as Anthracnose. These observations indicate that visually similar symptoms remain a source of classification error even when the overall validation accuracy is high.

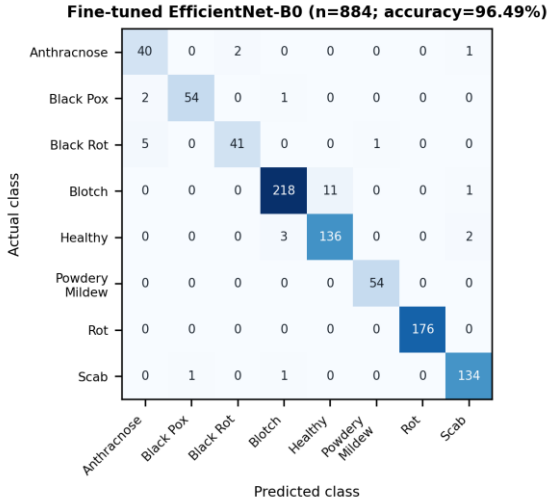


Fig. 3. Confusion matrix of fine-tuned EfficientNet-B0 on 884 validation crops.

E. Application Integration and Practical Interpretation

The Flask prototype shows how the two-stage model can be integrated with an interactive diagnosis workflow. The detector locates the regions of fruit within the image and the classifier generates a disease label and confidence score, given an image uploaded or captured by the user. This type of integration may be helpful in initial screening and may facilitate planning of observations for subsequent investigation. It is not a substitute for a plant pathologist's expertise in judging a tree's disease status, nor a guarantee of benefits to orchards when the treatment is applied.

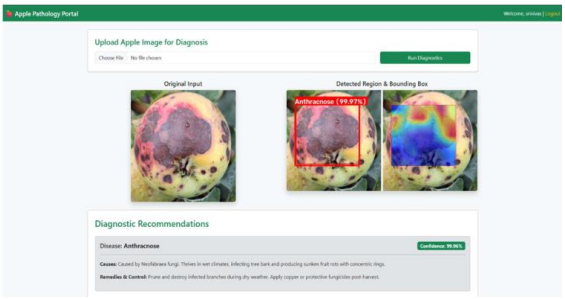


Fig. 4. Example output of the flask prototype of fruit localization, diagnosis and a visual heat overlay.

F. Limitations

The study reported here has a number of limitations. The data set was first collected from two Roboflow repositories, and the URLs of the repositories, versions of the data sets, and the terms of the licence should be documented to ensure the experiment was completely reproducible. Second, the reported classification result is a validation result; there is no independent test set collected from orchards at different locations, seasons, cultivars and devices reported. Third, the thesis points out light sensitivity and weighty occlusion as limitations. Fourth, the 3.0 ms is an inference of the detector. The end to end latency measurement didn't include the classifier, the image transfer and ROI extraction, nor the application response. Lastly, you'll measure the performance on an RTX 2060 workstation, and deploy to mobile or edge hardware separately requires profiling and model optimization.

The next step in validation should involve using images collected independently from the orchards and document the split at the source-image level to avoid crosses between the different crops in the orchard, measure full-pipeline latency, and evaluate the model on mobile or edge devices. The class-wise errors indicate that more examples and specifically more visual examples of similar symptoms, for Black Rot and Blotch might be helpful.

V. CONCLUSION

This paper presented a two-stage hybrid framework for apple fruit disease detection in complex field images. Fine-tuned YOLO11n was used to localize fruit regions, and EfficientNet-B0 was used to classify the extracted regions into eight health and disease categories. The fine-tuned detector achieved 94.22% precision, 92.60% recall, 96.86% mAP@0.50 and 78.63% mAP@0.50:0.95, with a Stage 1 inference latency of 3.0 ms per image on an RTX 2060. EfficientNet-B0 achieved 95.14% baseline validation accuracy and 96.49% after fine-tuning with class-weighted cross-entropy. The results support the use of separate models for fruit localization and fine-grained diagnosis. However, independent field testing and end-to-end timing are needed before making claims about generalization or real-time deployment. The framework provides a basis for further evaluation on diverse orchard data and resource-constrained devices.

REFERENCES

[1] Tian Y, Li E, Liang Z, Tan M and He X (2021), "Diagnosis of Typical Apple Diseases: A Deep Learning Method Based on Multi-Scale Dense Classification Network", Frontiers in Plant Science, Vol. 12, Article 698474. https://doi.org/10.3389/fpls.2021.698474

[2] Azgomi H, Roshannia Haredasht F and Safari Motlagh M R (2023), "Diagnosis of some apple fruit diseases by using image processing and artificial neural network", Food Control, Vol. 145, Article 109484. https://doi.org/10.1016/j.foodcont.2022.109484

[3] Gao L, Cao H, Zou H and Wu H (2025), "DMN-YOLO: A Robust YOLOv11 Model for Detecting Apple Leaf Diseases in Complex Field Conditions", Agriculture, Vol. 15, No. 11, Article 1138. https://doi.org/10.3390/agriculture15111138

[4] Leiva F, Gabioud Rebeaud S and Christen D (2026), "Real-time identification and quantification of apple scab on fruit in preharvest and postharvest conditions using YOLO11: a deep learning approach", Plant Methods, Vol. 22, Article 29. https://doi.org/10.1186/s13007-026-01508-7

[5] Tan M and Le Q V (2019), "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", Proceedings of the 36th International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 97, pp. 6105-6114.

[6] He K, Zhang X, Ren S and Sun J (2016), "Deep Residual Learning for Image Recognition", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[7] Huang G, Liu Z, van der Maaten L and Weinberger K Q (2017), "Densely Connected Convolutional Networks", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4700-4708.

[8] Ultralytics (2024), "Ultralytics YOLO11", Ultralytics documentation and software repository, available at: https://docs.ultralytics.com/models/yolo11/ (software version used: 8.4.94).

[9] Roboflow (n.d.), "Roboflow Universe", available at: https://universe.roboflow.com/. The exact two project URLs, dataset names, versions and licence details must be added by the authors before submission.

[10] Loshchilov I and Hutter F (2019), "Decoupled Weight Decay Regularization", International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1711.05101