# Sky View Aerial LandScape Classification

Chris Mei (*z5627083*) Chenlin Li (*z5527155*)
Bo Xu (*z5470301*) Mahshid Gohari (*z5560591*)
Xinpeng Zhang (*z5556103*)

***This project examines and compares multiple computer vision techniques for aviation scene classification on the SkyView dataset, with 15 balanced landscape classes. Traditional models as well as deep learning approaches are adopted. An example is feature-based techniques such as Scale-Invariant Feature Transform and Local Binary Pattern to extract features, while classification is done through logistic regression, K-Nearest Neighbour, Support Vector Machine, Random Decision Tree. Certain convolutional neural networks have been fine-tuned for end-to-end image classification such as EfficientNet, ResNet, and VGGNet. The artificial class imbalance introduced in data was also examined as well as some argumentation methods such as SMOTE, oversampling MixUp, and weighted training. Experimental results indicate that EfficientNetB0 performs best in terms of overall accuracy, while traditional pipelines can greatly improve by considering class balancing techniques. These results reflect the advantage that deep learning offers in image recognition complex tasks as well as highlight the importance of hybrid evaluation using classic methods for complete performance benchmark classification.***

***Keywords—Deep learning, Machine learning, Aerial Scene Classification, Feature extraction***

## I. Introduction

Currently, landscape classification plays an important application in urban planning, environmental monitoring, as well as in processes for responding to disasters, by allowing for quick, high-volume semantic labeling of drone and satellite data. This project classifies 15 landscape types ranging from Agriculture to Residential to River.

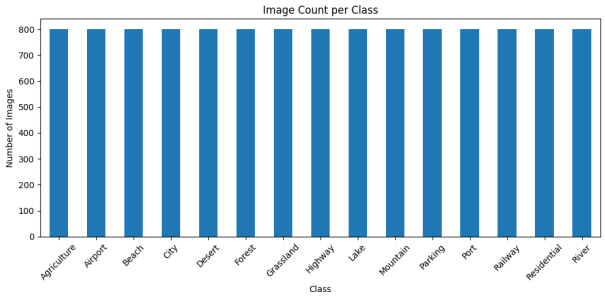
We benchmark two families of approaches: Traditional machine learning methods, employing SIFT with BoW and LBP feature extractors coupled with classifiers such as Logistic Regression, k-NN, SVM and Random Forest.

Modern deep learning architectures, using ResNet-18, EfficientNet-B0 and Vgg-16 works with some standard data augmentation methods such as MixUp, class-weighted loss, and two-stage retraining.

To assess robustness, we split the dataset into 80% for training and 20% for validation, we also simulated a long-tail imbalance dataset, with data-level preprocessing method such as SMOTE and MixUp, and algorithm-level class weighting and mitigation.

## II. Literature Review

The Aerial Landscape Dataset contains 15 classes, each initially with 800 images, each image has 256 *256 pixels.




### A. Tranditional feature based methods

Scale Invariant Feature Transform (SIFT) and Local Binary Patterns (LBP) are two local descriptors commonly employed in image classification. SIFT is capable of local key point detection that is invariant to scales, rotation, and variations in lighting, making it suitable for challenging image classification tasks [1]. Bag-of-Words builds a fixed-length feature vector by clustering key-point descriptors (with algorithms such as k-means), representing global image structure from the distribution of features in images.

LBP offers compact texture descriptions by encoding pixel intensity local patterns. It's fast computationally, light-weight invariant, and most suitable for texture-dense classification tasks [2].

### B. Deep learning architectures

Convolutional neural networks have been demonstrated to learn features in an automatic manner directly from raw pixels in images very effectively. But with an increase in depth, standard neural networks tend to suffer from vanishing gradients, making them hard to train.

Hence, we decide to use ResNet, it solves this problem by using residual (skip) connection:

$$Output = F(x) + x$$

Instead of learning the full mapping, the network learns the residual ($F(x)$), and adds the input $x$ back in. This makes it easier for the network to learn identity mappings, improving training and convergence.

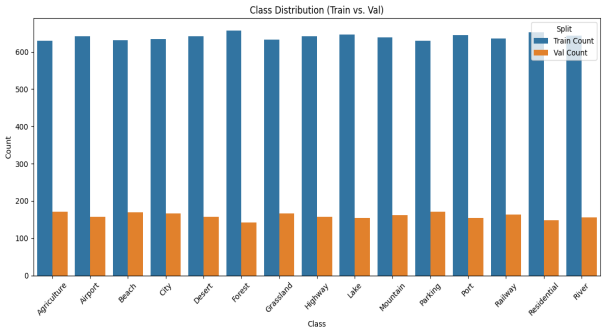
We also used EfficientNet-B0, an efficient yet powerful CNN architecture that enhances performance. Mobile reverse bottleneck convolution is applied, more efficient and smoother Swish activation is used compared to ReLU, and squeezed excitation block is added for focusing on relevant features. EfficientNet achieves high computational efficiency with high accuracy by virtue of these design points.

Lastly, we also have the classic, simple architecture-based deep convolutional model, VGG-16, with 3×3 convolutional layer stack followed by fully connected layers after max pooling. VGG-16 is more computationally complex but lacks some architecture features such as residuals or attention. However, despite not being innovative in architecture, it is still an incredibly strong image-classification baseline owing to being simple as well as deep [3].

## III. Methodology

### A. Data preparation

We organized the dataset into training data and validation sets based on predefined indexes to have uniform comparison across approaches. We have not used any form transformation like rotation, flipping, or color dithering, so that data is in an original state.




### B. Preprocessing

a) Feature based methods

For SIFT, Images were preprocessed by Gaussian Blur, Laplacian sharpening or CLAHE, and key-points were extracted using SIFT. These descriptors were clustered via Minibatch K-Means into visual words, resulting in a 100-dimensional BoW histogram per image.

And for LBP, each image was processed to extract uniform LBP features with 8 points around each pixel, resulting in compact histograms representing local texture patterns

b) Deep learning architectures

In EfficientNetB0, pixel value were normalized using its built-in method (preprocess_input). A pretrained EfficientNetB0 was used as a frozen feature extractor. Also, a custom classifier head was added. This setup is common in transfer learning, where the pretrained convolutional base captures generic visual patterns and only the classifier adapts to the task-specific data.

For ResNet-18, it has been pretrained on ImageNet, The final fully connected layer was modified to match the number of outputs which are 15 classes. and set the Adam optimizer with learning rate $1 \times 10^{-4}$.

### C. Evaluation

We use accuracy score, precision score, recall score and f1-score as the metric of comparison. Accuracy measures how many predictions your model got right over the total number of predictions. Precision measures how many of the predictions for a given class were actually correct. Recall measures how many of the actual class instances your model correctly identified. And we also use confusion matrix to visualize the prediction.

## IV. Result

### A. Feature methods

a) SIFT with BoW

Logistic regression on both SIFT and BoW gave high precision and accuracy. Logistic regression demonstrated robustness in learning diverse visual features for classification. The value of K-NN is marginally less in terms of metrics. This suggests sensitivity to localized feature distribution.

| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 63.25% | 0.63 | 0.63 | 0.63 |
| K-NN | 56.45% | 0.57 | 0.57 | 0.56 |
| Decision tree | 60.46% | 0.59 | 0.60 | 0.59 |
| SVM | 70.00% | 0.70 | 0.70 | 0.70 |

SVM performed better, with higher recall and precision results compared to those from logistic regression. This shows that there is high linear separability in the space of features generated from SIFT-BoW. k-NN performed reasonably but with less accuracy and precision due to sensitivity to local variation and inherently existing noises in the vocabulary words' representation.

b) LBP

| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 41.25% | 0.39 | 0.41 | 0.38 |
| K-NN | 51.41% | 0.52 | 0.51 | 0.51 |
| Decision tree | 59.41% | 0.58 | 0.59 | 0.58 |
| SVM | 54.31% | 0.54 | 0.54 | 0.53 |

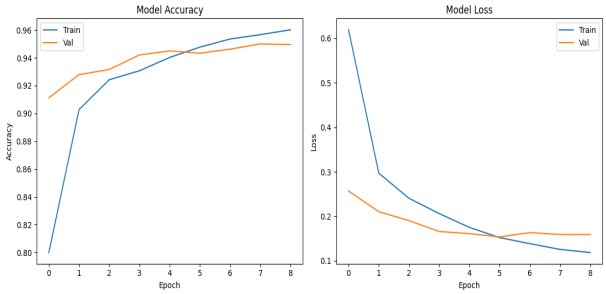
Random decision trees perform better than SVM. As the results on SVM, overall accuracy is less compared to SIFT-BoW. This finding reflects limitations of texture-based local descriptors alone in classifying complex images. Performance by k-NN is very similar to SVM, suggesting that local texture features by LBP alone can perform neighbourhood-based classification effectively.

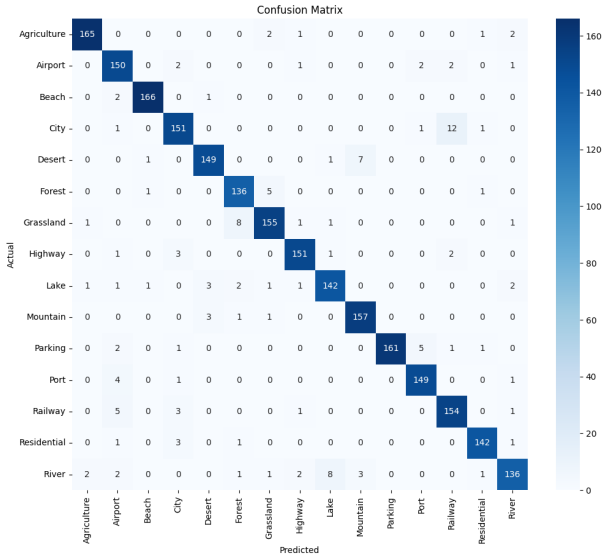
### B. Deep learning architectures

a) EfficientNetB0

The overall accuracy is 94% with high single class performance. The f1 for all classes is more than 90%, reflecting good performance in generalization. Recall for "*river*" is marginally less (0.87) because it is visually very close to '*lake*' or "*harbor*" in the aerial view. No data augmentation has been utilized in this experiment,

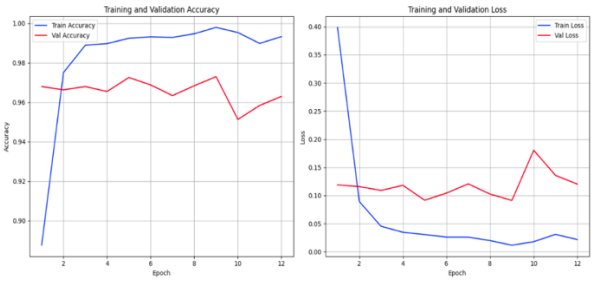
reflecting the high baseline performance of pre-trained EfficientNetB0 on this data.



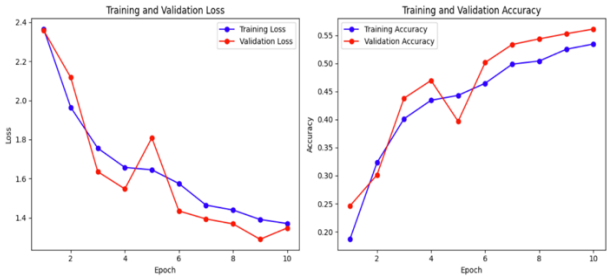

b) ResNet-18

The model achieved 96.88% accuracy with a macro F1-score of 0.9688.




c) VGG-16

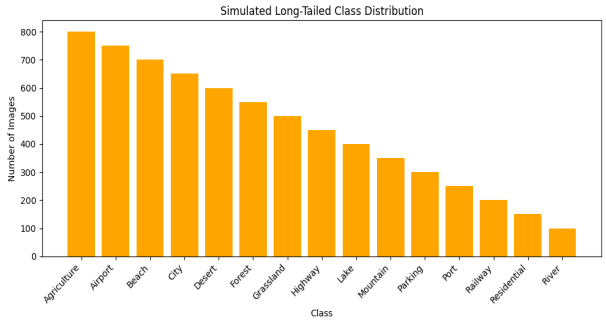
The VGG-16 model fine-tuned by data augmentation techniques achieves a peak validation accuracy of 56.7%, while the final training accuracy reaches about 52.4%. Beyond 10 epochs, the loss curve steadily decreases, and the accuracy curve gradually increases, indicating that the learning is effective and steadily converges without overfitting.




## V. Further Research

***What if the dataset is resampled to make it an imbalanced classification problem?***

Firstly, we need to create an imbalanced dataset, we reduce the number of images progressively lower. A long-tailed class distribution was simulated by retaining decreasing amounts of data per class: from 800 down to 50 images. To avoid data leak, we obtain a subset from the dataset as the training set. Make sure it has this long-tailed distribution and obtain another subset from the remaining data as the testing data with the same imbalanced distribution.




### A. *Feature based methods*

SIFT with Bow (Imbalanced)

| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 48.00% | 0.62 | 0.48 | 0.50 |
| K-NN | 36.96% | 0.59 | 0.37 | 0.39 |
| Decision tree | 42.85% | 0.60 | 0.43 | 0.45 |
| SVM | 50.25% | 0.65 | 0.50 | 0.51 |

LBP (Imbalanced)

| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 33.02% | 0.43 | 0.33 | 0.31 |
| K-NN | 31.85% | 0.48 | 0.32 | 0.32 |
| Decision tree | 35.45% | 0.41 | 0.35 | 0.34 |
| SVM | 32.00% | 0.45 | 0.33 | 0.32 |

Because k-NN has no parameters. It only stores training data, and all computations are performed during prediction. The only way to deal with this situation is to enhance the training data itself. Therefore, we apply synthetic minority class oversampling technique (SMOTE) for preprocessing. It is a method of generating synthetic data points for minority classes instead of just copying the existing data points [4]. For each minority class sample, SMOTE selects one or more of its nearest neighbours (from the same class). It then creates a synthetic sample by inserting between the original point and its neighbour points. It balances the class distribution without losing most of the data.

For logistic regression, random decision tree and SVM we could set their parameter 'class_weight' to 'balanced'. It makes the model automatically adjusts weights inversely proportional to class frequencies in the input data. It gives higher penalty to errors on the minority classes and helps the model pay more attention to the underrepresented classes.

SIFT with BoW:

| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 53.67% | 0.61 | 0.54 | 0.55 |
| K-NN | 47.85% | 0.51 | 0.48 | 0.49 |
| Decision tree | 25.32% | 0.30 | 0.24 | 0.18 |
| SVM | 64.65% | 0.66 | 0.64 | 0.63 |

LBP:

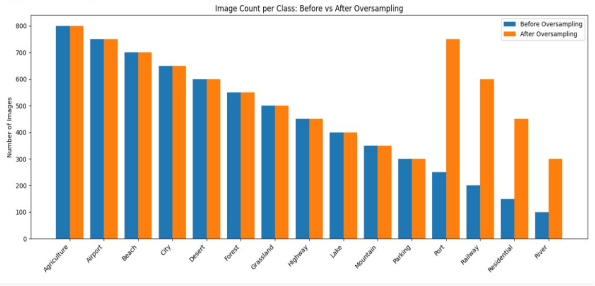
| Classifier | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| Logistic regression | 46.19% | 0.50 | 0.46 | 0.47 |
| K-NN | 39.31% | 0.45 | 0.39 | 0.40 |
| Decision tree | 50.21% | 0.54 | 0.50 | 0.49 |
| SVM | 21.84% | 0.20 | 0.22 | 0.14 |

From the results, we could know that class imbalance severely impairs model performance, especially on minority classes. And apply balancing techniques are crucial and effective—both data-level and algorithm-level.

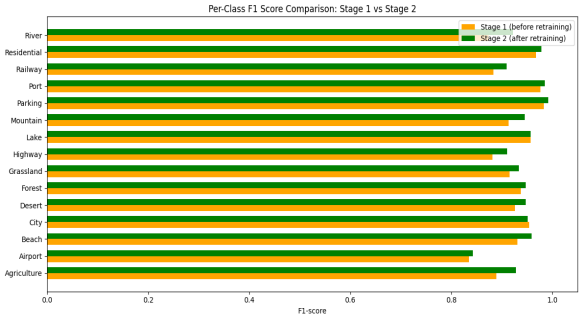
## B. *Deep learning architectures*

In deep learning, we choose to conduct this study on Efficient-B0. Here it is noted that it possesses good generalizing capability with validation accuracy higher at all times compared to training accuracy, indicating less overfitting. While performance in dominant classes is good, some few classes suffer from declining f1 -score, pointing towards the data unbalanced training.

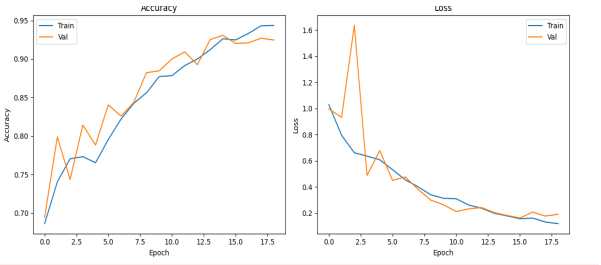
Therefore, to counteract this imbalance, we located the few classes that were represented by less than 250 images, and we applied two data augmentation techniques on their images: horizontal mirroring and contrast adjustment. These augmentations act to effectively augment the samples for all the most underrepresented classes such that more samples exist for the classifier to learn from. We plotted and compared the image counts for every class after and before oversampling after enhancing. This verifies that our enhancing approach managed to improve class distributions to an even balance without removing the intrinsic data bias.




Two-stage training was used to further tune the model. Stage 1 used fully end-to-end training for 30 epochs before being prematurely stopped to avoid overfitting. In stage 2, we left the model's feature extractor layer frozen but fine-tuned just the classifier head for an additional three epochs. This separately retrained approach enabled better generalization along with stronger decision boundaries for the minority class.



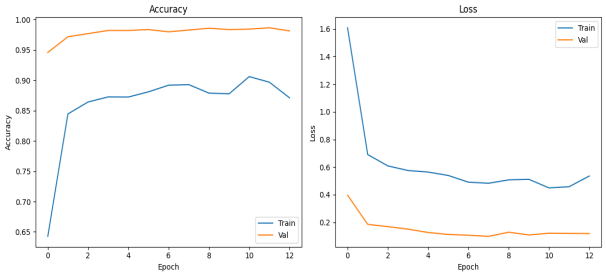

The proposed model attains an overall validation accuracy of 92.02%, which is quite a robust performance in all classes. Interestingly, f1 -scores for tail classes like rivers and residents are 0.87 and 0.97, respectively, affirming that oversampling with weighted training effectively enhances recognition of minority classes.




In addition, we try to use another data level argumentation method, MixUp argumentation method. It was integrated into the training pipeline. This method interpolates both the images and their labels by sampling a mixing coefficient from a Beta distribution. MixUp helps improve the smoothness of decision boundaries and acts as a regularizer, which is particularly valuable in low-data

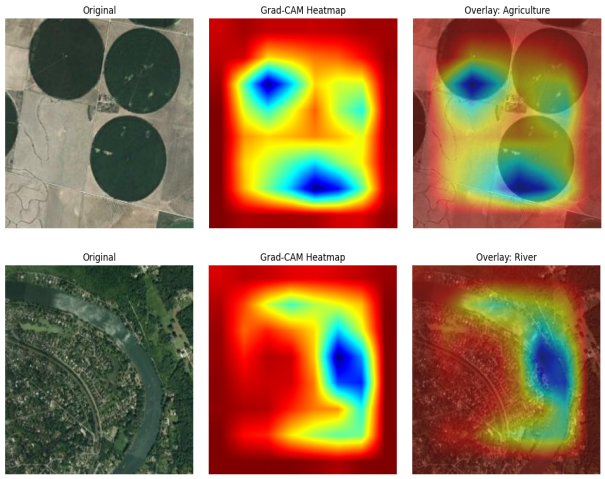
regimes, such as with minority classes in an imbalanced setting [5].

We also employed a two-stage training approach to further tune the network. In stage 1, we trained the full network end-to-end with MixUp as well as weighted loss. Early stopping with patience 5 epochs was applied to prevent overfitting. In stage 2, all convolutional layers were frozen, and we retrained just the end classification head for three more epochs. This permitted the classifier to reset its boundaries while keeping useful features learned in stage 1.




## C. *Grad-CAM Visualizations*

To demonstrate focus, we applied Grad-CAM to one high-frequency class (agriculture) and one low-frequency class (rivers). As we can see in the image, the model is able to point to regions of interest for both classes, for example, circular fields in agriculture and courses of water in rivers. This verifies that learning meaningful spatial patterns for classification even for sparse classes has been accomplished by the model.

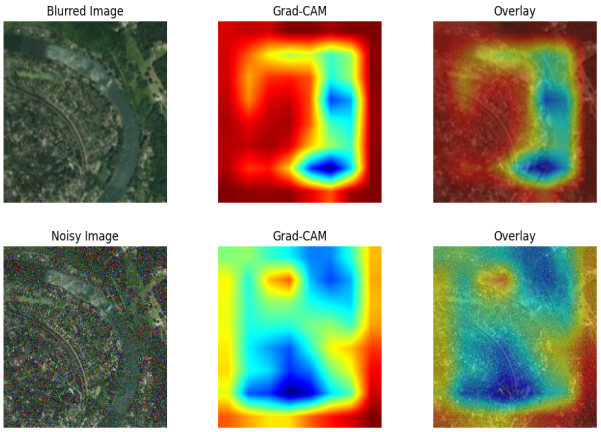



To further explain the model predictions and evaluate robustness under visual disturbances, Grad-CAM was used to generate visual explanations for both clean and interference inputs.

We selected a correctly classified image from the '*River*' class and applied the following perturbations:

- Gaussian Blur: to simulate minor visual distortion or low-quality sensors
- Occlusion: to test the effect of missing or obstructed image regions
- Gaussian Noise: to simulate sensor noise or poor lighting conditions

Every image (original and perturbed) was filtered through the EfficientNet-B0 model it had been trained on, overlaid on top with the resulting Grad-CAM heatmap to emphasize areas that play most importance in the model's choice.




These visualizations prove that not only does the model identify classes under perfect conditions but also preserves some degree of interpretability and reliability in perturbed settings. This is important in real-world aerial image use cases in which data quality can change based on atmospheric, sensor, or environmental factors.

# VI. Discussion

It could be observed that there have some Confusion Patterns such as *River* and *Lake*, *Mountain* and *Grassland*, *Forest* and *Grassland*, indicating limited discriminative power of fixed descriptors under context variation.

Data-level: Although SMOTE partially increased the performance on tailed class but cannot fully close the gap.

Algorithm-level: Class-weighted loss and two-stage fine-tuning effectively rebalance deep models, improving minority-class F1 by over 30 % compared to no mitigation.

Overall, deep learning approaches the superior accuracy, robustness and interpretability for aerial scene classification, especially under long-tail conditions, while traditional methods offer a lightweight alternative with acceptable performance in balanced settings.

# VII. Conclusion

In this work, an exhaustive comparison between conventional machine learning approaches and deep learning techniques is conducted to classify aerial scenes on the dataset SkyView. The main results include:

Higher accuracy of deep models: ResNet-18 and EfficientNet-B0 reach 98.8 % accuracy in balanced settings and retain > 98.5 % in simulated long-tail imbalance with MixUp, class-weighting, and two-stage training. Limitations of pipelines based on features: SIFT-BoW with SVM and LBP

with logistic regression perform moderately but decrease rapidly in response to imbalanced class distributions.

Efficient imbalance reduction: Data-level techniques must be combined with algorithm-level ideas to salvage minority class performance particularly in deep models. Interpretability observations: Confusion matrices and visualization via Grad-CAM affirm that high-level semantic cues are better represented by deeper features that avoid visually similar group confusions.

## REFERENCES


[1] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60(2), 91-110

[2] Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 971-987

[3] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. International Conference on Learning Representations (ICLR 2015)

[4] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357

[5] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. International Conference on Learning Representations (ICLR 2018)