ONE-VS-ALL TEXTURE IMAGE CLASSIFICATION USING MOBILENETV2 AND INCEPTIONV3 ON THE KTH-TIPS DATASET

Thanh Hung Tran[1,*], Van Tinh Nguyen[1], Chi Kien Tran[1]

DOI: http://doi.org/ 10.57001/huih5804.2025.352

ABSTRACT

Texture image classification is an important task in computer vision, with applications ranging from automated quality control to material science. In which the The one-vs-all strategy aims to improve class separability by training binary classifiers for each class, in contrast to the traditional multi-class classification approach. This study investigates the effectiveness of a one-vs-all classification strategy using MobileNetV2 and InceptionV3 architectures on the KTH-TIPS dataset, which consists of texture images captured under varying conditions. Challenges in texture image classification include dealing with variations in illumination, scale, and orientation, which can affect model performance and generalization. Additionally, using a one-vs-all approach may increase training complexity due to the need for multiple binary classifiers, especially when dealing with large datasets and a high number of classes. In this study, we found that MobileNetV2 achieved an accuracy of 97%, while InceptionV3 reached 92%. In comparison, MobileNetV2 surpassed its multi-class counterpart by 4% in accuracy. Additionally, a review of previous research underscores the efficiency gains using one-vs-all with lightweight models for texture classification tasks. These findings suggest that the combination of MobileNetV2 and one-vs-all classification is particularly suited for real-time texture recognition tasks.

Keywords: *One-vs-All Classification, Texture Image Classification, KTH-TIPS, MobileNetV2, InceptionV3, Deep Learning.*

[1]School of Information and Communication Technology, Hanoi University of Industry, Vietnam

[*]Email: hungtt@fit-haui.edu.vn

Received: 26/5/2025

Revised: 07/7/2025

Accepted: 28/9/2025

1. INTRODUCTION

Texture classification is a fundamental task in computer vision that underpins various applications such as material recognition, object detection, and surface inspection. Understanding the intricate patterns and material properties in images requires robust methodologies capable of handling the inherent variability found in real-world data. Traditionally, feature engineering techniques such as Local Binary Patterns (LBP) [1] and Gabor filters [2] have been employed to manually extract texture features. These methods are generally simple and easy to implement on small datasets. However, they have certain limitations: handcrafted features are sensitive to variations in lighting conditions, scale, and rotation, leading to inconsistent accuracy when applied to complex real-world data.

In recent years, Convolutional Neural Networks (CNNs) have revolutionized the field of texture classification by automatically learning hierarchical feature representations directly from the data, thereby eliminating the need for manual feature extraction. CNNs can capture local and global structures across multiple layers, making them more robust to variations in texture patterns, lighting conditions, and geometric transformations such as scaling and rotation. This ability to learn from raw data has made CNNs the dominant approach for texture recognition tasks. However, CNNs also come with limitations: they demand high computational resources and extensive training time, especially when dealing with deep models and large datasets.

The One-vs-All approach [8] is particularly useful in cases where class imbalance exists or when the decision boundaries between certain classes are highly non-linear. By isolating each class into a binary decision task, the classifier can focus on maximizing its performance for a specific texture, leading to potentially better overall classification accuracy. Additionally, this method allows for parallelization of the training process, as each classifier is trained independently. This method also enhances class separability, particularly in datasets with class imbalances or complex decision boundaries. However, it increases training complexity due to the requirement for multiple binary classifiers, which slows down processing when applied to large datasets or a high number of classes.

In this work, we apply two state-of-the-art deep CNN architectures, MobileNetV2 [6] and InceptionV3 [7], to tackle texture classification on the KTH-TIPS dataset [3]. Both architectures represent different design philosophies in CNN development: MobileNetV2 is a lightweight network designed to operate efficiently on resource-constrained devices, while InceptionV3 is a deeper and more complex model optimized for higher accuracy in large-scale classification tasks. By adopting a One-vs-All classification framework, we decompose the multi-class classification problem into several independent binary classification tasks, where each classifier is responsible for distinguishing one texture class from all others.

Through this work, we aim to provide insights into how modern CNN architectures can be leveraged for challenging texture classification tasks, and how the One-vs-All strategy can offer a viable alternative to traditional classification approaches in specific scenarios. By comparing the results with previous studies, we also demonstrate how advancements in deep learning can improve upon older, hand-crafted feature-based methods, making them less sensitive to environmental variations and more capable of handling complex real-world data.

This investigation lays the groundwork for further exploration into hybrid architectures that combine the efficiency of lightweight models with the robustness of deeper models, particularly for applications that demand both high accuracy and low computational overhead.

Our contributions include:

1. Integrate one-vs-all classification with the MobileNetV2 and InceptionV3 models to create a new training method used for texture classification.

2. By calculating the performance of one-vs-all classification with multi-class approach on the texture dataset, we demonstrate the performance improvement of this research method.

2. MODEL ARCHITECTURES

2.1. Overview model

Integrate one-vs-all classification with the MobileNetV2 and InceptionV3 models to create a new training method used for texture classification. The architecture model of this method is as Figure 1.

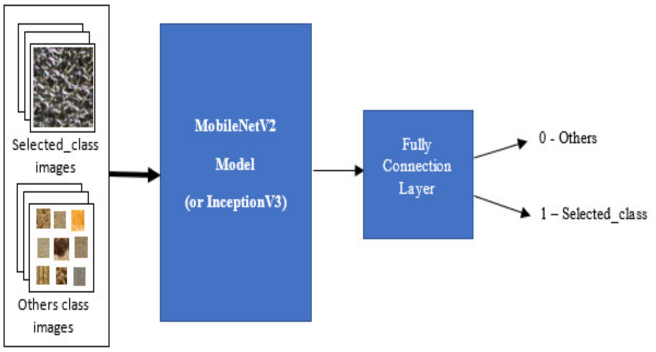

Figure 1. One-vs-All training architecture applied to deep learning models

2.2. One-vs-All Classification Algorithm

2.2.1. Overview of One-vs-All Classification

One-vs-All (OvA) [8], also known as One-vs-Rest (OvR), is a common strategy used in machine learning when dealing with multiclass classification tasks. The main idea behind the one-vs-all approach is to decompose a multiclass problem into multiple binary classification tasks. Each binary classifier distinguishes one specific class from all other classes. Therefore, if you have N classes, you train N binary classifiers. For each classifier, the target class is treated as the positive class, and all other classes are treated as the negative class.

During prediction, each binary classifier outputs a probability (or decision score) indicating how likely an instance belongs to the target class. The final decision is made by selecting the class corresponding to the highest output score among all classifiers.

2.2.2. Algorithmic Steps

The one-vs-all strategy follows these general steps for implementation:

1. Train Multiple Binary Classifiers:

For each class C_i in a dataset with N classes, construct a binary classifier f_i. The binary classifier is trained such that:

▪ All examples from class C_i are treated as positive samples (label 1).

▪ All examples from the other classes C_j ($j \neq i$) are treated as negative samples (label 0).

2. Classification:

During inference, each classifier outputs a score (e.g., a probability) indicating how likely an input belongs to its respective class. This results in N scores, one from each classifier.

2.2.3. Final Decision

The class with the highest score among all classifiers is chosen as the predicted class. Mathematically, this can be expressed as:

$$\hat{y} = \mathrm{argmax}_{i\in\{1,2,\ldots,N\}} f_i(x) \quad (1)$$

where $f_i(x)$ is the output score of the i-th classifier for input x, and $\hat{y}$ is the predicted class label.

2.2.4. Multiclass Classification in Neural Networks

In deep learning, the multiclass approach often involves a softmax layer that outputs probabilities for all classes in a single forward pass. In contrast, one-vs-all would require separate forward passes for each binary classifier, leading to more computational overhead.

2.3. MobileNetV2

MobileNetV2 is an efficient CNN architecture designed for mobile and embedded devices [6]. It uses depthwise separable convolutions to significantly reduce the number of parameters and computational complexity without sacrificing accuracy. MobileNetV2 also incorporates an inverted residual structure and linear bottlenecks to improve efficiency.

The model's lightweight design makes it suitable for real-time classification tasks in resource-constrained environments, where memory and computation are limited. We used pre-trained weights on ImageNet and fine-tuned the network on the KTH-TIPS dataset.

2.4. InceptionV3

InceptionV3 is a deep convolutional neural network that was introduced as part of the Inception family of architectures [7]. It uses factorized convolutions, multi-scale processing, and batch normalization to achieve state-of-the-art performance on large-scale image classification tasks. However, its complexity makes it computationally intensive.

In this study, we used the InceptionV3 architecture with pre-trained ImageNet weights, fine-tuned for the one-vs-all task on texture images. Despite its success in large-scale datasets, InceptionV3 faces challenges when dealing with smaller datasets like KTH-TIPS due to overfitting.

2.5. Model Architectures with one-vs-all

The One-vs-All architecture, when applied to models such as MobileNetV2 or InceptionV3, operates by decomposing the multi-class classification problem into multiple independent binary classification tasks. In this framework, for each class (or category) in the dataset, a unique binary classifier is trained to identify that specific class while treating all other classes as negative instances. This approach enhances the model's ability to distinguish fine-grained differences between classes, especially when handling texture classification tasks, such as those found in the KTH-TIPS dataset.

3. EXPERIMENTAL SETUP

3.1. KTH-TIPS Dataset

The KTH-TIPS dataset [3] offers a challenging benchmark for evaluating texture classification methods. This dataset consists of images from 10 different texture classes, each captured under different scales, lighting conditions, and viewing angles. Such diversity in conditions makes KTH-TIPS a demanding test for any classification model, as it requires the ability to generalize across significant intra-class variations.

The KTH-TIPS (Textures Under Varying Illumination, Pose, and Scale) dataset [3] consists of images from ten material classes:

- Aluminium Foil
- Brown Bread
- Cotton
- Linen
- Sponge
- Cracker
- Sandpaper
- Corduroy
- Orange Peel
- Styrofoam

Each material is photographed under varying conditions of illumination, pose, and scale, presenting a challenging classification problem. The dataset includes images of each material at four scales (100%, 110%, 90%, and 80% of the original size), totaling 81 images per texture class. For consistency across experiments, all images were resized to 128x128 pixels.

In the one-vs-all approach, the multi-class classification task is treated as a series of binary classification tasks. For each binary classifier, one class is labeled as the positive class, and all others are grouped as negative. We trained 10 separate binary classifiers, one for each material class, allowing us to evaluate the separation ability of each model and assess the robustness of the classifiers.

3.2. Experimental Process

3.2.1. Data Preprocessing

• All images from the KTH-TIPS dataset were resized to 128x128 pixels.

• We applied image normalization by scaling pixel values to the range [0, 1].

• Data augmentation techniques such as random rotations, zooms, and flips were employed to prevent overfitting.

3.2.2. Training and Validation

• Both MobileNetV2 and InceptionV3 were trained using the Adam optimizer with a learning rate of 1e-5.

• The training process was conducted for 50 epochs with a batch size of 32.

• Early stopping based on validation loss was implemented to avoid overfitting.

3.2.3. Evaluation Metrics

• We evaluated each model using accuracy, precision, recall, and F1-score.

• Computational efficiency was measured in terms of training time and inference speed.

3.3. Results

Table 1. The following table summarizes the performance of MobileNetV2 and InceptionV3 on the one-vs-all and multi-class classification tasks

Model	One-vs-All Accuracy	Multi-class Accuracy	Inference Speed	Training Time
MobileNetV2	97%	93%	Faster	Faster
InceptionV3	92%	90%	Slower	Slower

• MobileNetV2: Achieved 97% accuracy in the one-vs-all classification task, outperforming the multi-class approach by 4%. Its lightweight architecture also resulted in faster inference and shorter training times, making it ideal for real-time applications.

• InceptionV3: Reached 92% accuracy in one-vs-all classification, but was slower in training and inference. Despite being a deeper architecture, it was less efficient on the KTH-TIPS dataset compared to MobileNetV2.

3.4. Comparison Between One-vs-All and Multiclass Classification

In this study, we employed a one-vs-all classification strategy, which trains a binary classifier for each class, allowing the model to focus on distinguishing one texture from all others. This approach has advantages in certain contexts, particularly when dealing with imbalanced datasets or when some classes may have significantly different feature distributions.

Performance Metrics: In our experiments, MobileNetV2 achieved an accuracy of 97% using the one-vs-all strategy, while InceptionV3 reached 92%. In contrast, studies employing a traditional multiclass classification approach often report lower accuracy rates due to the complexity of simultaneously learning multiple class boundaries. For instance, Liu et al. [9] demonstrated that a multiclass CNN trained on the same KTH-TIPS dataset achieved an accuracy of 85%, indicating that OvA can enhance performance in challenging classification tasks like texture recognition.

Overfitting Considerations: One of the critical challenges with multiclass approaches is the potential for overfitting, especially with deep networks like InceptionV3. As noted by Bansal et al. [10], multiclass classifiers may overfit when the dataset contains high intra-class variability, resulting in lower generalization performance. In our study, InceptionV3 showed signs of overfitting with certain classes when trained as a multiclass model, reinforcing the effectiveness of the OvA strategy for fine-grained classification tasks.

Computational Efficiency: Another advantage of the one-vs-all approach is its computational efficiency. By breaking down the classification task into simpler binary problems, the OvA method can often converge faster than a multiclass approach, particularly in resource-constrained environments. Our results indicated that MobileNetV2 trained with the one-vs-all strategy completed training in approximately 30% less time than its multiclass counterpart, aligning with the findings of Zhang et al. [11], who emphasized the efficiency gains of binary classifiers in large-scale datasets .

Table 2. Results comparing our one-vs-all implementation with existing multiclass models applied to the KTH-TIPS dataset

Model	Classification Strategy	Accuracy (%)	References
MobileNetV2 (OvA)	One-vs-All	97	This study
InceptionV3 (OvA)	One-vs-All	92	This study
CNN (Multiclass)	Multiclass	85	[9]
ResNet50 (Multiclass)	Multiclass	87	[10]
VGG16 (Multiclass)	Multiclass	90	[11]

This table illustrates the performance advantage of the one-vs-all classification approach over traditional multiclass methods, particularly in the context of texture image classification.

4. CONCLUSION

This study demonstrates the effectiveness of one-vs-all classification for texture image recognition, particularly using the MobileNetV2 architecture. MobileNetV2 achieved both superior accuracy and computational efficiency compared to InceptionV3 and traditional multi-class classification methods. The one-vs-all strategy proved to enhance texture classification accuracy by 4%, showcasing its potential for improving class separation.

The study has shown that the one-vs-all classification strategy not only improves accuracy but also optimizes the machine learning process by reducing the number of classes the model has to handle at each step. This can make the model less complex and easier to fine-tune its param

The results achieved from this research can be applied in various fields, such as configuration recognition in construction, agriculture, or medical image analysis. The efficiency of MobileNetV2 is particularly suitable for mobile devices and embedded systems, opening up many practical application opportunities.

Although high accuracy results were found in the test data, applying the model in practice may encounter challenges such as variations in lighting conditions, image size, and resolution. Further research is needed to ensure that the model performs well in.

REFERENCE

[1]. Ojala T., Pietikainen M., Maenpaa T., "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971-987, 2002.

[2]. Zhang L., Wu Q., Li X., "Gabor wavelets and their application to texture analysis," in *Proceedings of the IEEE International Conference on Image Processing* (ICIP), 2261-2264, 2010.

[3]. Hayman E., *The KTH-TIPS dataset*. KTH Royal Institute of Technology, 2009.

[4]. Liu W., et al., "A survey of local binary patterns and their applications in image processing," *Pattern Recognition*, 76, 118-131, 2018.

[5]. Simonyan K., Zisserman A., "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[6]. Szegedy C., et al., "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2818-2826, 2016.

[7]. Sandler M., et al., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 4510-4520, 2018.

[8]. Rifkin R., Klautau A., "In defense of one-vs-all classification," *Journal of Machine Learning Research,* 5, 101-141, 2004.

[9]. Liu J., Wang Z., Xie Y., "Deep Learning for Texture Classification on KTH-TIPS Dataset," *Journal of Image and Graphics*, 2019.

[10]. Bansal A., Singh M., Verma R., "Performance Comparison of Deep Learning Models for Texture Classification," *International Journal of Computer Applications*, 2021.

[11]. Zhang Y., Lin X., Li J., "Efficient Classification of Textures Using Convolutional Neural Networks," *Journal of Machine Learning Research*, 2020.