applied sciences

MDPI

Article

Entropy-Based Ensemble of Convolutional Neural Networks for Clothes Texture Pattern Recognition

Reham Al-Majed and Muhammad Hussain *

Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; 442203765@student.ksu.edu.sa
* Correspondence: mhussain@ksu.edu.sa

Abstract: Automatic clothes pattern recognition is important to assist visually impaired people and for real-world applications such as e-commerce or personal fashion recommendation systems, and it has attracted increased interest from researchers. It is a challenging texture classification problem in that even images of the same texture class expose a high degree of intraclass variations. Moreover, images of clothes patterns may be taken in an unconstrained illumination environment. Machine learning methods proposed for this problem mostly rely on handcrafted features and traditional classification methods. The research works that utilize the deep learning approach result in poor recognition performance. We propose a deep learning method based on an ensemble of convolutional neural networks where feature engineering is not required while extracting robust local and global features of clothes patterns. The ensemble classifier employs a pre-trained ResNet50 with a non-local (NL) block, a squeeze-and-excitation (SE) block, and a coordinate attention (CA) block as base learners. To fuse the individual decisions of the base learners, we introduce a simple and effective fusing technique based on entropy voting, which incorporates the uncertainties in the decisions of base learners. We validate the proposed method on benchmark datasets for clothes patterns that have six categories: solid, striped, checkered, dotted, zigzag, and floral. The proposed method achieves promising results for limited computational and data resources. In terms of accuracy, it achieves 98.18% for the GoogleClothingDataset and 96.03% for the CCYN dataset.

Keywords: texture recognition; clothes pattern; deep learning; CNN; ensemble learning; entropy

Citation: Al-Majed, R.; Hussain, M. Entropy-Based Ensemble of Convolutional Neural Networks for Clothes Texture Pattern Recognition. *Appl. Sci.* **2024**, *14*, 10730. https://doi.org/10.3390/app142210730

Academic Editor: Andrea Prati

Received: 3 October 2024
Revised: 14 November 2024
Accepted: 18 November 2024
Published: 20 November 2024

1. Introduction

In recent years, the increased number of e-commerce platforms and online fashion retailers such as clothing recommendation systems, inventory management, and fashion trend analysis tackle vast amounts of image data in the textile and clothing field. This requires robust solutions for recognizing and retrieving clothes with specific properties, using machine learning algorithms to extract semantic information from images instead of relying on manual annotations. From another perspective, by virtue of the information explosion in this era, raw data captured by mobile devices and fed into powerful computation do not only facilitate daily life but can also increase the independence and confidentiality of disabled persons. Choosing clothes with appropriate designs and patterns is one of the challenging tasks for visually impaired people. Tackling this issue with an automatic assistive technique instead of family help would be preferred. Generally, automated clothing pattern recognition can significantly enhance efficiency in industrial applications and improve personalized fashion experiences.

Generally, as stated in [1], the texture of an image is identified by the variance in the spatial distribution of pixel intensities. As a result, texture classification is the task of identifying and recognizing this pattern distribution of pixels. To some extent, the texture classification task is similar to the object recognition task in that the strong correlation of pixel intensities is determined. However, the pixel arrangement in a small portion of

an image is repeated within the whole image, and thus the pattern of the image can be recognized even with small portions. On the other hand, in object recognition, the entire object should appear for accurate recognition.

Determining the pattern of clothes is a texture classification task where repeated basic primitives, such as dots, stripes, plaids, floral, etc., characterize the pattern. Figure 1 shows some popular patterns in our clothes. In the context of the computer vision field, extracting the pattern of clothing from natural photos is a challenging process due to variations in lighting, angles, and zoom if taken as selfies or from advertising boards on city street compared to professional photos.

Figure 1. Examples of six classes of clothes patterns: checkered, floral, dotted, solid, striped, and zigzag.

Moreover, images of clothing patterns contain a high degree of intraclass variations for the same pattern category compared to traditional texture images such as grass, wood, leaves, etc., shown in popular texture datasets such as Brodatz [2] that express less intraclass and intensity variation.

Traditional texture descriptors in the literature extract useful information related to image texture and are represented as a feature vector before classification. Local Binary Pattern (LBP) [3], Gray-Level Co-occurrence Matrices (GLCMs) [4], and Gabor filter banks are some of the most relevant texture descriptors that are widely used, especially with insufficiency in resources. Recently, deep learning methods, usually using convolutional neural networks (CNNs), have outperformed traditional techniques. CNNs are composed of convolutional, pooling, and fully connected layers and learn hierarchical features using convolutional layers at different levels; lower layers extract low-level features, and higher-level layers learn semantics. They have become popular due to their remarkably good performance in a wide range of visual recognition tasks, such as object recognition, compared to handcrafted features.

This work proposes a solution based on CNN models for the problem of clothes patterns. Usually, there is a small-scale dataset for clothes patterns and limited computational resources, which are not enough to train a deep CNN model and avoid the problem of overfitting. To overcome these issues, we propose an ensemble classifier that uses a pre-trained ResNet50 with non-local block (NL), a squeeze-and-excitation (SE) block, and a coordinate attention (CA) block as base learners. The predictions of base learners are fused using a novel fusion technique based on entropy voting. Motivated by the observation that the texture is defined as a set of repeated primitives along the whole picture, we propose using a deep learning method that uses a ResNet50 [5] pre-trained model on a massive dataset incorporating different blocks that embed global spatial information to capture channel interdependencies and long-range dependencies to classify clothes pattern into six categories: solid, striped, checkered, dotted, zigzag, and floral. Two methods for voting are evaluated for the final prediction by the combined learners: max voting and entropy voting. Moreover, different models are evaluated in ensemble learning, such as pre-trained DenseNet, EfficientNet, and ResNet152 models. Two popular methods are employed to reduce overfitting: data augmentation and dropout.

The main contributions of the presented research work are as follows:

- A deep learning-based ensemble classifier is created that uses CNN models as base learners and entropy voting as a fusion technique to recognize the clothes patterns. To tackle the problem of small datasets and overfitting, the pre-trained CNN models are adopted as base learners, which are fine-tuned to the clothes patterns.
- A simple and robust entropy voting-based fusion method is presented that fuses the decisions taken by the base learners, taking into consideration their uncertainties in decision making.
- The method is validated thoroughly using benchmark datasets for clothes pattern recognition.

The rest of the paper is organized as follows. Section 2 gives an overview of the state-of-the-art method proposed for clothes patterns. The proposed method is presented in Section 3. Experimental results are given in Section 4 and discussed in Section 5. Finally, Section 6 concludes the paper.

2. Related Works

Recently, there has been increased interest in the e-commerce fashion industry. Researchers have addressed several clothes-related problems such as clothing classification [6–9], attribute recognition [10–12], and fashion recommendation systems [13]. Some works have addressed specific attribute recognition such as fiber identification [14,15]. Others classify clothes images according to their pattern (texture) attribute [16–19].

As per image or texture recognition techniques, clothing pattern recognition methods are categorized in two main groups. First are traditional methods that adopt handcrafted feature extraction methods to extract useful features and then classify them using some classifiers such as Support Vector Machine (SVM), neural networks (NNs), or other classifiers. Moreover, texture feature extraction methods can either extract local features, such as Scale-Invariant Feature Transformation (SIFT), Speeded Up Robust Features (SURFs), and Histogram of Oriented Gradient (HOG), or global features, such as Radon Signature, Discrete Wavelet Transform (DWT), Gray-Level Co-occurrence Matrix (GLCM), and Local Binary Pattern (LBP). Yang et al. [16] proposed systems that used SIFT, Radon Signature, and DWT to extract local and global features. However, in Manisha's [17] work, SURF was used to capture local features and the global features were captured using DWT and GLCM, while both systems fed the extracted features into the SVM classifier. Loke [18] also utilized GLCM to extract global features but fed them to a random forest classifier. The second approach is the deep learning approach, which has been applied successfully in texture recognition in general and for clothes specifically [14,19]. Lee Stearns et al. [19] classified clothes images into one of six common patterns: solid, striped, checkered, dotted, zigzag, and floral. They adopted the state-of-the-art convolutional neural network model (ResNet-101 [5]) that was pre-trained on an ImageNet dataset [20]. Using a standard transfer learning approach, they fixed all layers except for the final densely connected classification layer and trained the weights for that layer using their dataset. Tena et al. [21] proposed a modified convolutional neural network (MCNN) that employs a random tuning strategy to optimize hyperparameters, specifically tailored to traditional Indonesian woven fabrics. It demonstrates superior performance compared to several pre-trained CNN models. Kumar et al. [22] proposed a network that employs LSTM to handle the complexities of sequential patterns in fabric textures, resulting in improved fabric texture classification and defect detection. The proposed method in [23] used a video-based fabric pattern recognition approach with a Bayesian-optimized CNN (Bayes Opt-CNN). Video streams of fabric surfaces are used to extract spatiotemporal features, while Bayesian optimization selects the best hyperparameters to improve the accuracy of pattern recognition. The work in [24] proposed a dual-branch network that incorporates local features extracted from a pre-trained ResNet 50 model with global features extracted by Residual Pooling Transformer (RPT) for general texture recognition.

Most of the recent works that utilize deep learning approach identify clothes patterns as one of several attributes resulting in poor pattern recognition accuracy. However, most of the works carried out from the perspective of clothes patterns specifically relied on handcrafted features combined with a machine learning classifier.

This work propose a clothes pattern recognizer that can be part of an assistive tool for visually impaired people. It is an ensemble classifier that constitutes three sub-models: a pre-trained ResNet50 model combined with a non-local block (NL), a squeeze-and-excitation (SE) block, and a coordinate attention (CA) block where the voting is based on entropy.

3. The Proposed Method

In this section, firstly, the problem of clothes pattern recognition is defined as a supervised classification problem. The proposed approach based on CNN models and entropy voting is then presented in detail.

3.1. Problem Definition and Formulation

The clothes pattern recognition problem is concerned with identifying the pattern of a textile captured under different constraints. An image of a fabric is given, and its pattern should be determined automatically according to predefined patterns. It is a multi-class, single-label classification problem.

Let $X \subset \mathbb{R}^{m \times n}$ be a set of clothes pattern images, each represented as a matrix of size $m \times n$, and $Y = \{0, 1, ..., c\}$ be the set of predefined labels (patterns). The problem of identifying clothes patterns is to design a mapping $f : X \longrightarrow Y$ that predicts the label $y \in Y$ for an unknown pattern input (image) $x \in X$, i.e.,

$$f(x; \theta) = y \tag{1}$$

where θ are the learnable parameters. We validate the proposed method in two different scenarios. In the first scenario, the number of labels (patterns) is six: checkered, patternless, striped, zigzag, floral, and dotted (i.e., $y = 1, 2, ..., 6$). The other scenario consists of four labels: checkered, patternless, irregular, and striped (i.e., $y = 1, 2, 3, 4$).

3.2. Ensemble CNN Model

We model f as an ensemble of CNN models and use entropy voting-based fusion i.e.,

$$f(x; \theta) = \varepsilon(f_1(x; \theta_1), f_2(x; \theta_2)...f_n(x; \theta_n)) \tag{2}$$

where $f_i, i = 1, 2, ...n$ are base learners, $\theta_i, i = 1, 2, ...n$ are their learnable parameters, and ε is the fusion function. There are two possibilities to design the base learners. One approach is to learn diverse base learners using the same model but different training sets drawn with, e.g., bootstrap sampling. Another approach is to use diverse models but train with the same dataset. We follow the second approach and employ CNNs to model $f_i s$. An overview of the ensemble is shown in Figure 2.

A convolutional neural network (CNN) is a deep end-to-end learning model that shows attractive performance in the computer vision field. It is typically composed of three types of layers: convolution, pooling, and fully connected layers. As opposed to conventional algorithms, CNNs do not require a prior human-crafted feature extraction. Instead, hierarchical features are learned adaptively through the network, where low-level information is involved in the shallow layers and high-level information is embedded in deep layers.

The transfer learning approach allows one to exploit and leverage the knowledge a pre-trained model learned from a specific task to a related target task and dataset. As the pre-trained models are trained on massive datasets, e.g., ImageNet, applying them to tasks similar to the task at hand has been proven to result in improved performance with reduced computational cost in terms of time and dataset size. In this work, we employ pre-trained

CNN models as base learners. Transfer learning is utilized to fine-tune pre-trained models where the task-dependent fully connected layer is replaced by another one whose neurons are equal to the number of the classes of the current problem. More specifically, some of the final layers are replaced by new trainable layers while the early layers' learned weights are kept as initial parameters. This procedure is motivated by the observation that the early layers of a model seem to involve general features that can fit other tasks different than the original one while the features in the final layers are problem-specific. As the model is modified, it is retrained using the clothes pattern dataset where the early layers are frozen and the newly added trainable layers are learned. Fine-tuning is performed by unfreezing layers of the original model and training the model by continuing back-propagation using the new dataset. This retraining should be at a very low learning rate to prevent significant updates to the gradient and thus avoid overfitting while improving the performance.

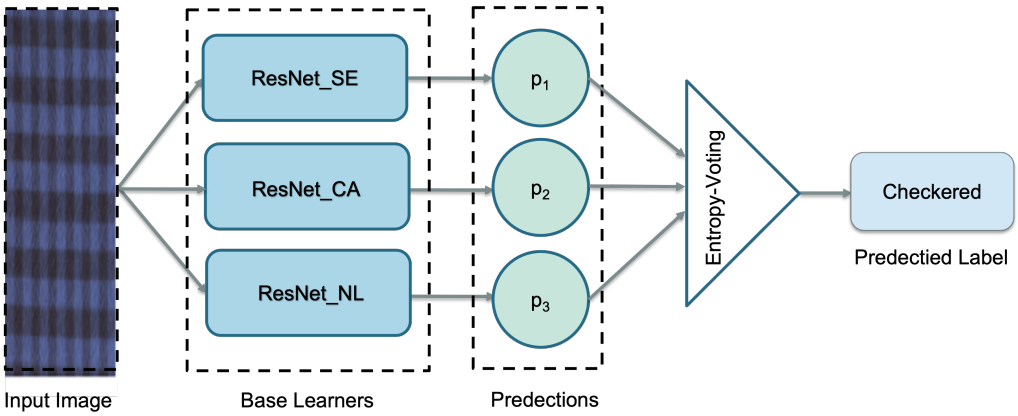


Figure 2. High-level depiction of the architecture of the proposed ensemble classifier.

ResNet50 is used as the backbone model in the transfer learning paradigm. The ResNet model [5] introduces a residual block where every few stacked convolution layers are skipped, and "shortcut connections" are initiated to bypass the layers in between. It addresses the problem of degrading accuracy with increased depth by reformulating the original unreferenced mapping into residual mapping. That is, the original mapping is reformulated into $F(x) + x$ as optimizing the residual function $F(x)$ is easier and capable of enhancing the performance even with increased depth. To perform the element-wise addition in the operation $F(x) + x$, x can be an identity mapping if it has the same dimension as $F(x)$ or can be linearly projected using a 1×1 convolution layer.

3.3. Base Learner CNN Models

We used the pre-trained ResNet50 as a backbone model and enhanced it to design three base models for the ensemble classifier by incorporating three blocks: a squeeze-and-excitation block (SE), coordinate attention block (CA), and non-local block (NL). The following paragraphs shed light on these blocks and the scheme of their inclusion within ResNet50.

ResNet50 is part of the ResNet family, proposed to tackle the vanishing gradient problem associated with training deep convolutional networks. It achieves high performance despite its relatively shallower depth compared to other types of ResNets, making it ideal for a wide range of applications. It consists of 50 layers organized into five groups where $ResG_i, i = 1, 2, ..., 5$ (see the detail given in Table 1 and shown in Figure 3). The first group $ResG_1$ consists of a 7×7 convolutional layer followed by a MaxPooling layer. The basic building block of the other four groups, $ResG_i, i = 2, 3, 4, 5$, is a bottleneck residual block as shown in Figure 4. $ResG_2$, $ResG_3$, $ResG_4$, and $ResG_5$ consist of three, four, six, and three bottleneck blocks. The stride of the last layer of the last block of each group is two, which downsamples the output of each group by a ration of two. The bottleneck architecture of each block ensures more expressive power with smaller number of learnable parameters.

Table 1. ResNet50 architecture, where fs, #f, s, mp, GAP, and FC stand for filter size, number of filters, stride, max pooling, global average pooling, and fully connected layer.

Group	Layer (fs, #f, s)	Input	Output
$ResG_1$	$(7 \times 7, 64, 2)$ $(3 \times 3, \text{mp}, 2)$	224×224	112×112
$ResG_2$	$\begin{bmatrix} (1 \times 1, 64, 1) \\ (3 \times 3, 64, 1) \\ (1 \times 1, 256, 1) \end{bmatrix} \times 3$	112×112	56×56
$ResG_3$	$\begin{bmatrix} (1 \times 1, 128, 1) \\ (3 \times 3, 128, 1) \\ (1 \times 1, 512, 1) \end{bmatrix} \times 4$	56×56	28×28
$ResG_4$	$\begin{bmatrix} (1 \times 1, 256, 1) \\ (3 \times 3, 256, 1) \\ (1 \times 1, 1024, 1) \end{bmatrix} \times 6$	28×28	14×14
$ResG_5$	$\begin{bmatrix} (1 \times 1, 512, 1) \\ (3 \times 3, 512, 1) \\ (1 \times 1, 2048, 1) \end{bmatrix} \times 3$	14×14	7×7
GAP			
FC+Softmax			

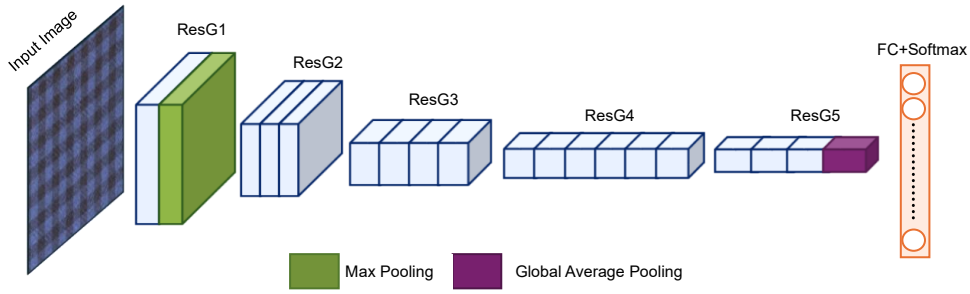


Figure 3. Detail of ResNet50 architecture.

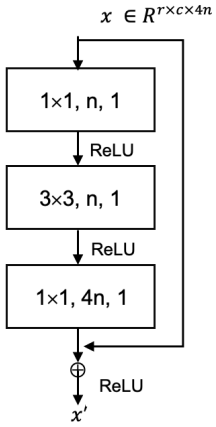


Figure 4. Architecture of bottleneck residual block.

3.3.1. ResNet50 with Squeeze-And-Excitation Block (ResNet_SE)

Squeeze-and-excitation (SE) block [25] improves the representation power of a CNN by modeling the interdependencies between the channels in two steps: squeezing and excitation. The squeeze step aggregates the feature maps across spatial dimensions to produce a channel descriptor using global average pooling. This is followed by an excitation operation, which learns the channel-wise importance of emphasizing the informative ones

and suppressing the less important ones by the simple gating mechanism of a bottleneck with two fully connected (FC) layers. The input feature maps are then re-weighted to generate the output of the SE block, which can then be fed directly into subsequent layers. We incorporate four SE blocks in a pre-trained ResNet50 model at the end of each stage, as shown in Figure 5.

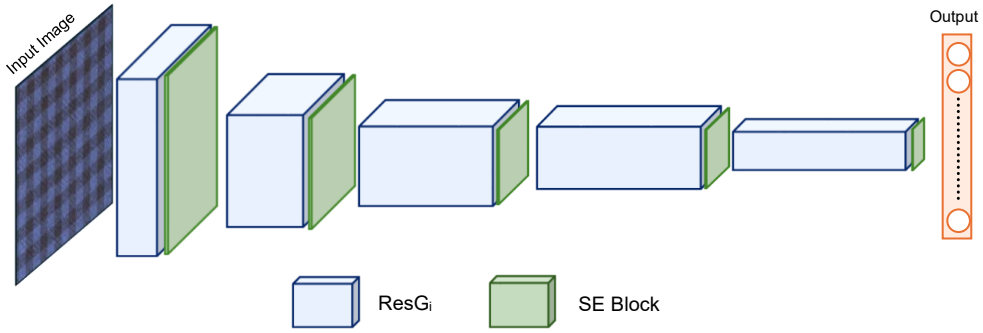


Figure 5. ResNet50 with SE Blocks. $ResG_i$ is the ith group of ResNet blocks.

3.3.2. ResNet50 with Coordinate Attention Block (ResNet_CA)

Channel attention has a considerable effect on improving network performance. Coordinate attention [26] is an attention mechanism that embeds positional information into channel attention. First, the channel attention is factorized into two parallel 1D poolings to aggregate features along the width and height spatial dimensions to capture long-range interactions spatially with precise positional information. In the second step, the coordinate attention is generated by 1×1 convolutional transformations. We designed base learner ResNet_CA by incorporating this block into the pre-trained backbone ResNet50 as shown in Figure 6.

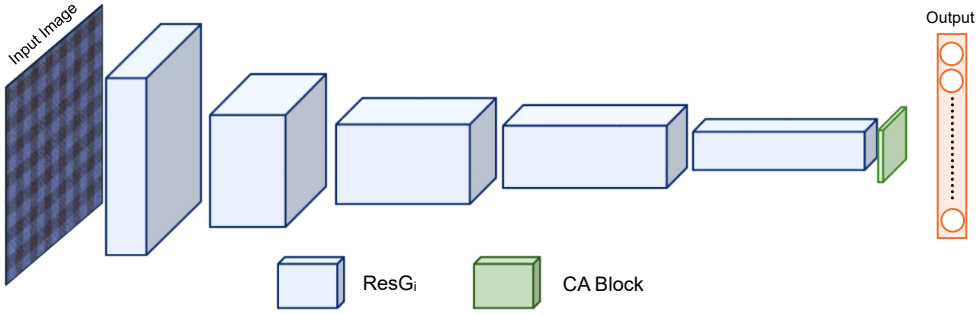


Figure 6. ResNet50 with CA block. $ResG_i$ is the ith group of ResNet blocks.

3.3.3. ResNet50 with Non-Local Block (ResNet_NL)

Non-local block [27] captures long-range dependencies directly and efficiently in one shot instead of repeated convolutional layers. The response at each position is the weighted sum of all other responses in that map. The non-local block is defined as

$$z_i = W_i y_i + x_i \tag{3}$$

$$y_i = \frac{1}{C(x)} \sum_{\forall j} f(x_i, x_j) g(x_j) \tag{4}$$

The unary function g in the above equation is a linear embedding that computes a response at a position. Even though there are several choices for the binary function f that computes the affinity between two responses, the enhanced performance of non-local models is attributed to the generic behavior of its operation, not to the choice of this function. To design ResNet_NL to be used in the proposed method, a non-local block is incorporated

in a pre-trained backbone ResNet50 before the last block of the third stage as shown in Figure 7.

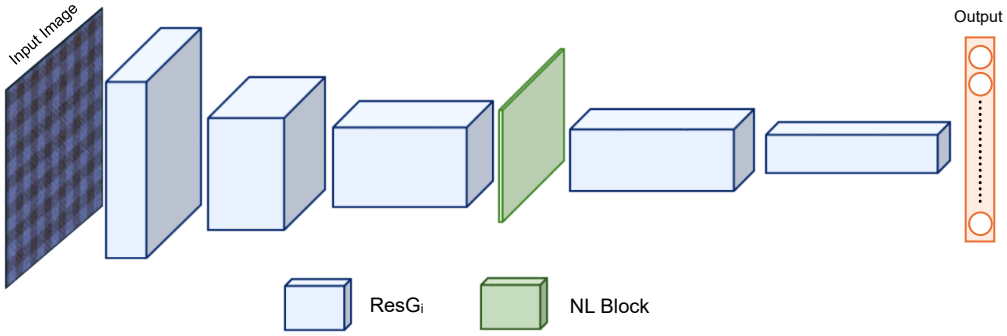


Figure 7. ResNet50 with NL block. $ResG_i$ is the ith group of ResNet blocks.

All three base models described above are diverse by design, with each incorporated block contributing an aspect of diversity. The SE block acts as a self-attention mechanism across channels, dynamically recalibrating each channel's importance to emphasize informative features and suppress less useful ones. On the other hand, the NL block captures contextual information across the entire input, making it particularly valuable for tasks requiring global context, such as pattern recognition. Meanwhile, the CA block embeds positional information into channel attention, aggregating features along two spatial directions, which enhances spatial sensitivity for tasks such as pattern recognition. Thus, incorporating these blocks provides rich contextual, spatial, and channel-wise attention capabilities, strengthening feature representations. They are trained in such a way that the incorporated block layers are learned while keeping the pre-trained ResNet50 layers frozen. After the initial training of each block, all layers (including ResNet50 layers) of each base model are kept unfrozen and fine-tuned with a very small learning rate.

3.4. Fusion Based on Entropy Voting

Each base model j is trained independently, yielding a predicted label y_j and a vector of probabilities $p_j = (p_j^1, p_j^2, ...p_j^c)$ which is fused for the final prediction of unseen patterns.

The base models are trained independently, and the predictions from these models are fused to determine the final prediction. The simplest and most commonly used fusion method is max voting (i.e., majority voting), where the label of a test sample is given according to the number of classifiers who vote for some specific label. The main drawback of this approach is that it does not consider the uncertainties in base learners' decisions. We propose a new fusion method in this work: entropy voting. Entropy measures the uncertainty in that the higher the entropy, the higher the uncertainty. For any unknown sample x, each base learner j computes the vector $p_j = [p_j^{1,x}, p_j^{2,x}, ..., p_j^{c,x}]^T$ of posterior probabilities belonging to each label. These probabilities are used to compute the entropy e_j^x for classifier j according to the following equation [28]:

$$e_j^x = \sum_{i=1}^{|C|} p_j^{i,x} log(p_j^{i,x}), j = 1, 2, ..., k \tag{5}$$

where C is the number of labels (classes) and $p_j^{i,x}$ is the probability that a test sample x belongs to class i predicted by classifier j, and k is the number of base learners.

The label of a classifier with the lowest entropy is chosen as the predicted label for a test sample as the decision of this classifier involves the lowest uncertainty. Formally, let e_j^x be the entropy calculated for a test sample x by the classifier j, then the set $E = \{e_j^x, \forall j \in 1...k\}$

represents the set of entropies for all k base learners of the ensemble. The predicted label assigned to the sample x by entropy voting is calculated as follows:

$$J = argmin_j\{e_j^x\} \tag{6}$$

and the predicted label of x is

$$\ell = \ell_J \tag{7}$$

$$\ell_J = argmax_C\{p_j^{1,x}, p_j^{2,x}, ..., p_j^{c,x}\} \tag{8}$$

The predicted label l_J is based on the base model, which has the least uncertainty and the highest level of confidence.

4. Experiments and Results

In this section, we provide details of the experimental setup and describe the datasets used to evaluate the performance of the proposed method. We then present the results of the experiments that were conducted to validate the effectiveness of the proposed method.

4.1. Experimental Setup

All layers of the core model (ResNet50) were frozen, and the weights of classification layers and the incorporated blocks (SE, CA, or NL) were initialized by the Lecun initializer. We used the Adam optimizer with a mini-batch size of 100 and a learning rate of 0.0001. The models were then fine-tuned where all layers were unfrozen, and the learning rate was set to 1×10^{-6}. The proposed system was implemented and evaluated in the Python programming language using the Keras library in the Google Colab cloud service. It was tested against two datasets. The first one was used in [19], was collected from Google, and contains 317–584 images per class (2764 images total) of size 224×224. The number of clothing pattern designs in this dataset is six: solid, striped, checkered, dotted, zigzag, and floral. The other dataset was CCYN [16], which includes 627 images of four clothing pattern designs: plaid, striped, patternless, and irregular, with 156, 157, 156, and 158 images in each category. The resolution of each image is 140×140. Both datasets were split into 60%, 20%, and 20% for training, validation, and testing, respectively. To overcome the overfitting problem due to insufficient training datasets, the simplest approach was to employ geometric transformations to generate additional samples. The geometric transformations specifications shown in Table 2 were applied.

Table 2. Geometric transformation.

Zoom Range	Rotation Range	Shear Range	Horizontal Flip
0.3	30	0.2	True

4.2. Experimental Results

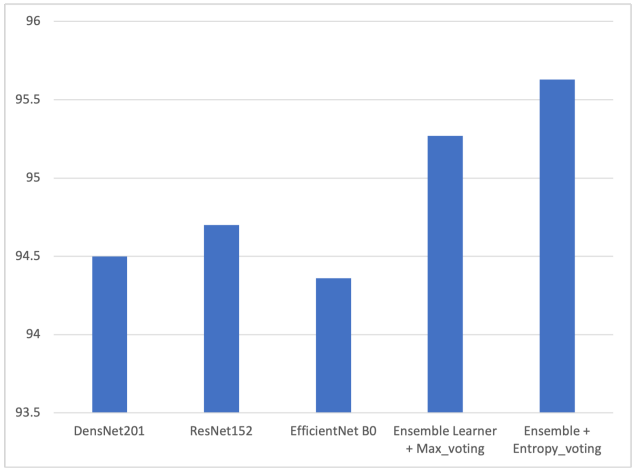
We conducted several experiments to examine the effect of the ensemble classifier by measuring the accuracy (Acc) on the test dataset. We considered widely used high-performance CNN architectures (ResNet152, DenseNet201, EfficientNetB0) as base learners to show the effectiveness of the proposed CNN architectures (ResNet50_CA, ResNet50_SE, ResNet50_NL) as base learners. Further, to show the effectiveness of our proposed fusion method, we performed experiments with max voting and entropy voting.

4.2.1. The Effect of Base Learners

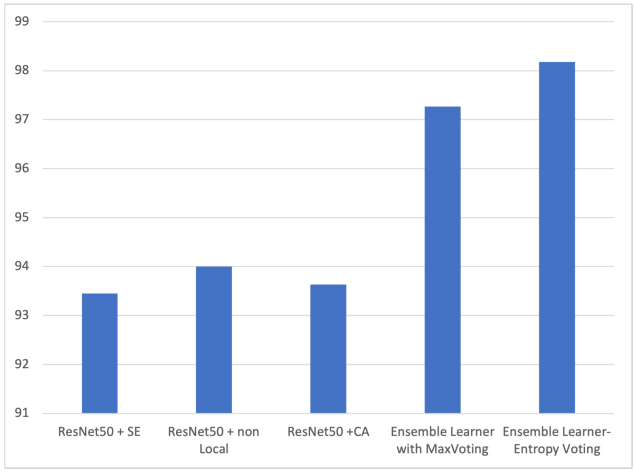
First, we performed experiments with individual CNN models; the results are shown in Table 3. The ResNet152, DenseNet201, and EfficientNetB0 models performed better than the ResNet50_CA, ResNet50_SE, and ResNet50_NL models.

Next, we performed experiments to show the effectiveness of the proposed ensemble classifier (ensemble classifier 2) in comparison with ensemble classifier 1 based on the

ResNet152, DenseNet201, and EfficientNetB0 models. Figure 8 and Table 3 show the performance of the individual models along with the performance of ensemble classifiers. As shown in the figure, the accuracy of individual models does not exceed 94%, while the ensemble classifiers show improvement in performance. Ensemble classifier 1, based on the three popular high-performance models (ResNet152, DenseNet101, and EfficientNetB0), enhances the accuracy by about 0.6%, as shown in Figure 8a.



(**a**)



(**b**)

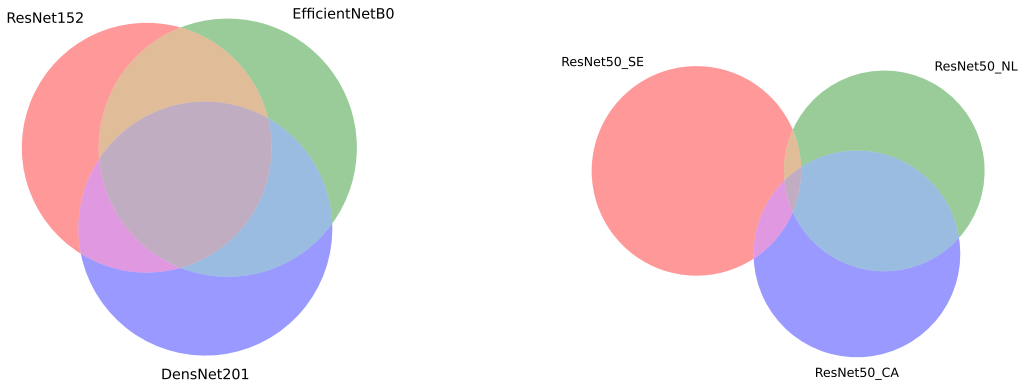
Figure 8. The performance of two ensemble classifiers. (**a**) The performance in terms of accuracy of ensemble learner 1 and the base learners. (**b**) The performance in terms of accuracy of ensemble learner 2 and the base learners.

However, ensemble classifier 2, which integrates the pre-trained RseNet50 with the SE attention block, RseNet50 with the CA block, and RseNet50 with the NL block, improves the accuracy up to 4%, as shown in Figure 8b. Though the performance of the base learners in the case of ensemble classifier 2 is not better than that of the ResNet152, DenseNet201, and EfficientNetB0 models, the performance of ensemble classifier 2 is better than that of ensemble classifier 1. This is in accordance with the theory of ensemble learning, which recommends the use of weak and diverse base learners. As the ResNet50_CA, ResNet50_SE, ResNet50_NL models exhibit weak and diverse performances, as shown in Table 3, ensemble classifier 2, based on these models, significantly outperforms ensemble classifier 1. The Venn diagrams in Figure 9 show the error analysis of each individual base learner. The non-overlapping areas represent samples misclassified by only one learner,

while the two-circle overlap indicates samples misclassified by two learners, with the possibility that the third learner correctly classifies them. A three-circle overlap signifies that all three learners failed to classify those samples. Figure 9b shows a slight overlap between the errors made by the base learners of ensemble classifier 2. While ResNet50 with the SE block has mostly unique errors, ResNet50 learners with the NL and CA blocks share about 50% of their errors. Analyzing the contributions of each classifier using the proposed entropy-based voting method, the base learners with the SE, NL, and CA blocks contributed to 23%, 47%, and 30% of the classifications, with error rates of 0.8%, 2.3%, and 1.8%, respectively. In contrast, Figure 9a shows a greater overlap between the errors of the base learners, indicating fewer opportunities for the ensemble classifier to improve performance on these samples. Furthermore, using the proposed entropy-based voting method, the base learners contribute with proportions of about 89%, 9%, and 2% for ResNet152, DenseNet, and EfficientNet, respectively. The dominance of ResNet152, combined with the increased overlap, explains the limited performance improvement of ensemble classifier 1.

Table 3. Results summary for the ensemble learners (ELs).

Method	Accuracy
ResNet152	94.7%
DensNet201	94.5%
EfficientNetB0	94.36%
Ensemble classifier 1 + Max voting	95.27%
Ensemble classifier 1 + Entropy voting	95.63%
ResNet50_CA	93.63%
ResNet50_SE	93.45%
ResNet50_NL	94%
Ensemble classifier 2 + Max voting	97.27%
Ensemble classifier 2 + Entropy voting	98.18%



(**a**) (**b**)

Figure 9. Venn diagram of base learners' errors. (**a**) Error analysis of base learners of ensemble classifier 1. (**b**) Error analysis of base learners of ensemble classifier 2.

4.2.2. The Effect of Fusion Techniques

Further, we performed experiments to show the effectiveness of the proposed fusion method. The rightmost two bars in Figure 8a,b compare the ensemble classifiers using two different voting methods: max voting and entropy voting. The max voting shows less enhancement in performance for the ensemble classifier; the reason is that it does not consider how much the model is confident about its decision. The superior performance of entropy voting is due to the fact that it takes into account the certainty of a model in its classification decision.

4.2.3. The Analysis of the Performance of the Ensemble Classifier

Figure 10 shows the confusion matrix of the ensemble classifier based on the base learners: RseNet50 with the SE attention block, RseNet50 with the CA attention block, and RseNet50 with the NL block. This matrix shows that the proposed method correctly classifies almost all classes; there are just a few misclassifications. All checkered and floral patterns are correctly classified. Only 1% of striped patterns are misclassified as checkered, floral, solid, and zigzag each. Further, 1% and 2% of zigzag patterns are misclassified as striped and floral, respectively. About 2% of dotted patterns are misclassified as solid and striped each. Only 1% of solid patterns are incorrectly classified as zigzag.

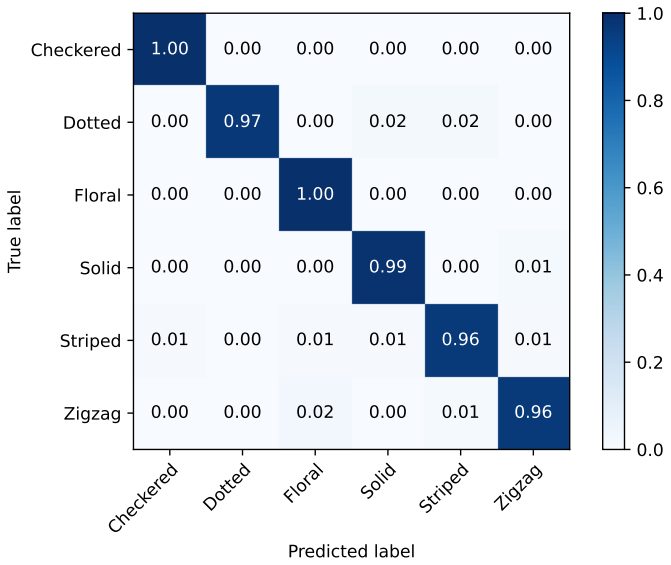


Figure 10. Confusion matrix showing the decision making of the ensemble classifier.

5. Discussion

The end-to-end deep learning technique used in this work proves its effectiveness in capturing discriminative features outperforming the hand-engineered features. As shown in Table 4, the entropy-based ensemble learner shows a significant improvement compared with most related works that rely on a single clothes attribute: its pattern. It shows its robustness as it generalizes well and gives good performance when used on two different datasets.

Table 4. Comparison with state-of-the-art works.

Dataset	Ref	Method	Accuracy
CCYN [16]	Yang et al. [16]	(SIFT+STA+RanonSig) + SVM	92.55 %
	Manisha [17]	(SURF+GLCM+DWT) + SVM	76.25%
		CNN–Ensemble learner	96.03%
Collected from Google [19]	Stearns et al. [19]	CNN (ResNet101)–Transfer learning	91.7%
		CNN–Ensemble learner	98.18%

The experimental results show the effectiveness of the fused learning algorithms in comparison with using each one separately. This improvement in performance definitely requires that each individual classifier is performing well. Incorporating different blocks with the pre-trained model (ResNet50) as a participating model resulted in diverse models, each of which extracts pattern features from different angles, yielding more reliable complementary prediction. Analyzing each pattern (label) classification performance in each individual model in Figure 11, it can be deduced that the classifiers are diverse in a way that each classifier tackles the data differently as they perform differently in some

patterns. For example, "solid"-labeled clothes are classified much better in the model that incorporates the non-local (NL) block than the one that incorporate coordinate attention (CA). This could be because CA embedded positional information which can be confused for "solid"-labeled clothes since, even they are solid, the type of the fabric itself may affect the prediction and be considered as checkered or dotted, such as in some kinds of fabrics like knitted wool or linen. Furthermore, it can be shown that ResNet50 incorporated with the CA block performs the best among the other models in capturing "stripes" and "zigzag" patterns, which span the whole shot by their nature.

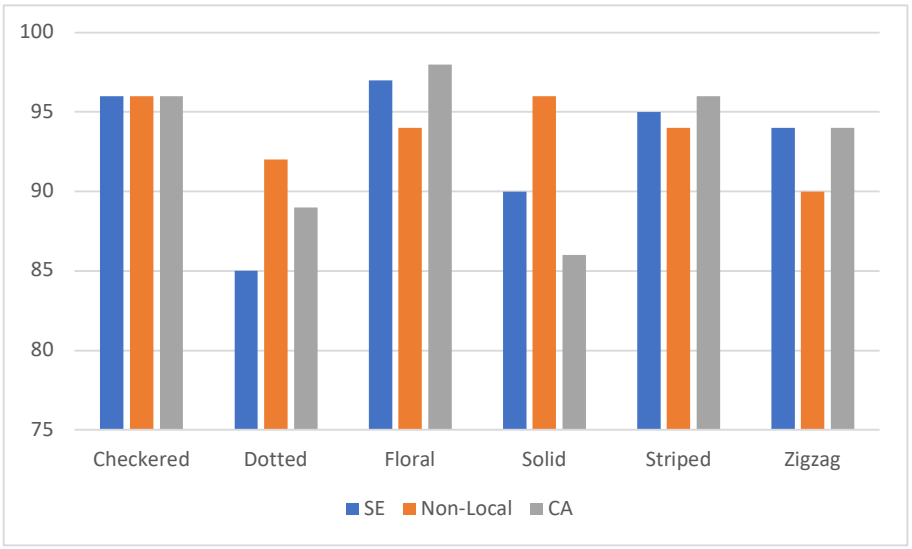


Figure 11. Performance of the base learners for each class.

As per any deep learning method, the crucial consideration for success is the volume and diversity of the training dataset. This work can be improved using a more efficient approach to augment training data, such as active learning, used in [28], where the learner can determine what the instances will be trained on according to some 'worthiness' indicator. Moreover, a massive clothes dataset with comprehensive annotations used for multi-attribute clothes recognition such as DeepFashion [8] can be used to train the proposed system. However, in this work, it is only tested against available single-attribute works' datasets.

6. Conclusions

This paper presented an ensemble classifier based on CNN models as base learners and entropy voting as a fusion technique for clothes pattern classification. To evaluate the usefulness of the proposed method, we examined two design approaches: an ensemble classifier based on high-performance complex CNN models (ResNet152, DenseNet101, and EfficientNetB0) and an ensemble classifier based on custom-designed simple variants of ResNet50, incorporating building blocks such as NL block, SE block, and CA block. Also, to show the effectiveness of the proposed new fusion method, two fusion methods were evaluated: one that depends on the popular majority voting technique and is called max voting and a new proposed one that depends on the certainty of a model to label a new test sample, referred to as entropy voting. The results indicate that the second ensemble approach, where the variants of ResNet50 are used as base learners and entropy voting is used for fusion, outperforms the other approaches.

Author Contributions: Conceptualization, R.A.-M. and M.H.; methodology, R.A.-M.; software, R.A.-M.; validation, R.A.-M. and M.H.; formal analysis, R.A.-M.; investigation, R.A.-M.; resources, R.A.-M.; data curation, R.A.-M.; writing—original draft preparation, R.A.-M.; writing—review and editing, M.H.; visualization, R.A.-M.; supervision, M.H.; project administration, M.H.; funding acquisition, M.H. All authors have read and agreed to the published version of the manuscript.

Funding: The research was supported under Researchers Supporting Project number RSP2024R109, King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This study used a dataset in the public domain. [CCYN] [https://doi.org/10.1109/THMS.2014.2302814].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cavalin, P.; Oliveira, L.S. A review of texture classification methods and databases. In Proceedings of the 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), Niteroi, Brazil, 17–18 October 2017; pp. 1–8.
2. Brodatz, P. *Textures; a Photographic Album for Artists and Designers*; Dover Publications: New York, NY, USA, 1966; p. xiv, 112p.
3. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
4. Haralick, R.M.; Shanmugam, K.; Dinstein, I.H. Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **1973**, *SMC-3*, 610–621. [CrossRef]
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
6. Rajput, P.S.; Aneja, S. IndoFashion: Apparel classification for Indian ethnic clothes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 3935–3939.
7. Yu, F.; Du, C.; Hua, A.; Jiang, M.; Wei, X.; Peng, T.; Hu, X. EnCaps: Clothing image classification based on enhanced capsule network. *Appl. Sci.* **2021**, *11*, 11024. [CrossRef]
8. Liu, Z.; Luo, P.; Qiu, S.; Wang, X.; Tang, X. DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations. In Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
9. Li, Z.; Sun, Y.; Wang, F.; Liu, Q. Convolutional neural networks for clothes categories. In Proceedings of the Computer Vision: CCF Chinese Conference, CCCV 2015, Xi'an, China, 18–20 September 2015; Proceedings, Part II; Springer: Berlin/Heidelberg, Germany, 2015; pp. 120–129.
10. Lee, W.; Jo, S.; Lee, H.; Kim, J.; Noh, M.; Kim, Y.S. Clothing attribute extraction using convolutional neural networks. In Proceedings of the Knowledge Management and Acquisition for Intelligent Systems: 15th Pacific Rim Knowledge Acquisition Workshop, PKAW 2018, Nanjing, China, 28–29 August 2018; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2018; pp. 241–250.
11. Zhang, Y.; Zhang, P.; Yuan, C.; Wang, Z. Texture and shape biased two-stream networks for clothing classification and attribute recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13538–13547.
12. Xiang, J.; Dong, T.; Pan, R.; Gao, W. Clothing attribute recognition based on RCNN framework using L-Softmax loss. *IEEE Access* **2020**, *8*, 48299–48313. [CrossRef]
13. Shin, Y.G.; Yeo, Y.J.; Sagong, M.C.; Ji, S.W.; Ko, S.J. Deep fashion recommendation system with style feature decomposition. In Proceedings of the 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 8–11 September 2019; pp. 301–305.
14. Ohi, A.Q.; Mridha, M.F.; Hamid, M.A.; Monowar, M.M.; Kateb, F.A. Fabricnet: A fiber recognition architecture using ensemble convnets. *IEEE Access* **2021**, *9*, 13224–13236. [CrossRef]
15. Sonawane, C.; Singh, D.P.; Sharma, R.; Nigam, A.; Bhavsar, A. Fabric classification and matching using CNN and siamese network for E-commerce. In Proceedings of the Computer Analysis of Images and Patterns: 18th International Conference, CAIP 2019, Salerno, Italy, 3–5 September 2019; Proceedings, Part II 18; Springer: Berlin/Heidelberg, Germany, 2019, pp. 193–205.
16. Yang, X.; Yuan, S.; Tian, Y. Assistive clothing pattern recognition for visually impaired people. *IEEE Trans. Hum.-Mach. Syst.* **2014**, *44*, 234–243. [CrossRef]
17. Dhongade, M. Clothing pattern recognition for blind using surf and combined glcm, wavelet. *Int. J. Sci. Res. (IJSR)* **2015**, *2013*, 2319–7064.
18. Loke, K. Automatic recognition of clothes pattern and motifs empowering online fashion shopping. In Proceedings of the 2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW), Taipei, Taiwan, 12–14 June 2017; pp. 375–376.
19. Stearns, L.; Findlater, L.; Froehlich, J.E. Applying transfer learning to recognize clothing patterns using a finger-mounted camera. In Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility, Galway, Ireland, 22–24 October 2018; pp. 349–351.
20. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]

21. Tena, S.; Hartanto, R.; Ardiyanto, I. Content-based image retrieval for traditional Indonesian woven fabric images using a modified convolutional neural network method. *J. Imaging* **2023**, *9*, 165. [CrossRef] [PubMed]
22. Kumar, K.S.; Bai, M.R. LSTM based texture classification and defect detection in a fabric. *Meas. Sens.* **2023**, *26*, 100603. [CrossRef]
23. Mamun, A.A.; Nabi, M.; Islam, F.; Bappy, M.M.; Uddin, M.A.; Hossain, M.S.; Talukder, A. Streamline video-based automatic fabric pattern recognition using Bayesian-optimized convolutional neural network. *J. Text. Inst.* **2024**, *115*, 1878–1891. [CrossRef]
24. Liu, Y.; Dong, H.; Wang, G.; Chen, C. Dual-Branch Network based on Transformer for Texture Recognition. *Digit. Signal Process.* **2024**, *153*, 104612. [CrossRef]
25. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the EEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
26. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
27. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.
28. Zhou, Z.; Shin, J.Y.; Gurudu, S.R.; Gotway, M.B.; Liang, J. AFT*: Integrating active learning and transfer learning to reduce annotation efforts. *arXiv* **2018**, arXiv:1802.00912.