



## Article

# Real-Time Automated Classification of Sky Conditions Using Deep Learning and Edge Computing

Joby M. Prince Czarnecki \* , Sathishkumar Samiappan , Meilun Zhou, Cary Daniel McCraine and Louis L. Wasson

Geosystems Research Institute, Mississippi State University, Mississippi State, MS 39762, USA; Sathish@gri.msstate.edu (S.S.); mz271@msstate.edu (M.Z.); cdm366@msstate.edu (C.D.M.); LWasson@gri.msstate.edu (L.L.W.)

\* Correspondence: joby.czarnecki@msstate.edu

**Abstract:** The radiometric quality of remotely sensed imagery is crucial for precision agriculture applications because estimations of plant health rely on the underlying quality. Sky conditions, and specifically shadowing from clouds, are critical determinants in the quality of images that can be obtained from low-altitude sensing platforms. In this work, we first compare common deep learning approaches to classify sky conditions with regard to cloud shadows in agricultural fields using a visible spectrum camera. We then develop an artificial-intelligence-based edge computing system to fully automate the classification process. Training data consisting of 100 oblique angle images of the sky were provided to a convolutional neural network and two deep residual neural networks (ResNet18 and ResNet34) to facilitate learning two classes, namely (1) good image quality expected, and (2) degraded image quality expected. The expectation of quality stemmed from the sky condition (i.e., density, coverage, and thickness of clouds) present at the time of the image capture. These networks were tested using a set of 13,000 images. Our results demonstrated that ResNet18 and ResNet34 classifiers produced better classification accuracy when compared to a convolutional neural network classifier. The best overall accuracy was obtained by ResNet34, which was 92% accurate, with a Kappa statistic of 0.77. These results demonstrate a low-cost solution to quality control for future autonomous farming systems that will operate without human intervention and supervision.

**Keywords:** autonomous systems; cloud detection; low-altitude remote sensing; ResNet; UAS image quality



**Citation:** Czarnecki, J.M.P.; Samiappan, S.; Zhou, M.; McCraine, C.D.; Wasson, L.L. Real-Time Automated Classification of Sky Conditions Using Deep Learning and Edge Computing. *Remote Sens.* **2021**, *13*, 3859. <https://doi.org/10.3390/rs13193859>

Academic Editors: Karem Chokmani, Yacine Bouroubi and Saeid Homayouni

Received: 11 August 2021  
Accepted: 23 September 2021  
Published: 27 September 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



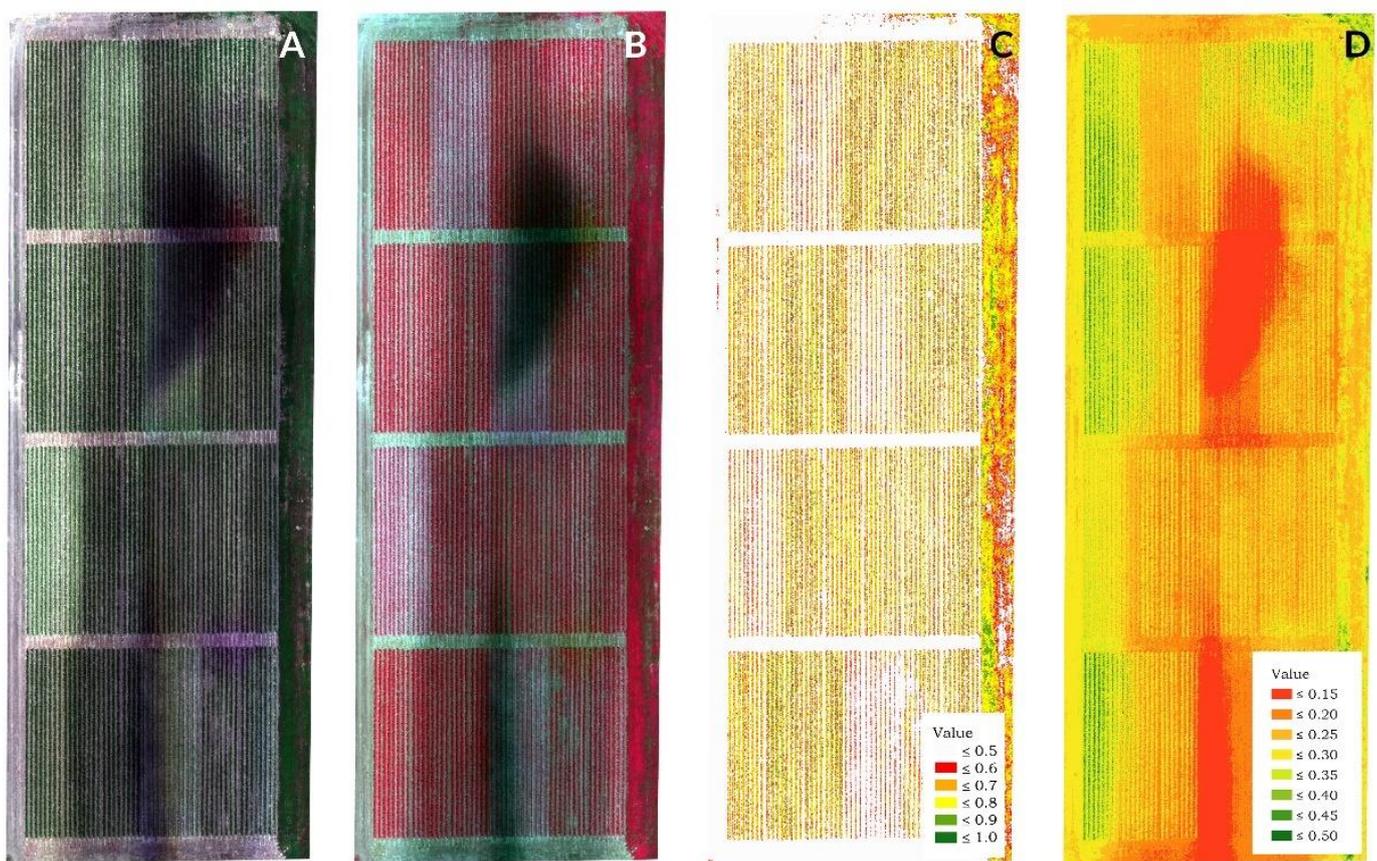
**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Remotely sensed imagery collected from low-altitude remote sensing platforms is frequently employed to identify spatio-temporal variability, and by extension, provides a basis for prescriptive interventions within agricultural production fields. In many areas, but especially in sub-tropical humid environments such as Mississippi, clouds have historically been, and continue to be, a challenge for remote sensing in agriculture [1–5]. With satellite imagery, clouds block the field of view, obscuring the area of interest, and making the imagery unsuitable for many applications. Eberhardt et al. [6] analyzed the potential to acquire cloud-free imagery over tropical and sub-tropical areas of Brazil using the occurrence of clear sky conditions over a 14-year period obtained from the MODIS cloud mask product. In some areas, clear skies were available only 15% of the time during the growing season. Ju and Roy [7] estimated the availability of sequential cloud-free Landsat ETM+ images (i.e., having a time series of images) in season at 60%.

The flexibility and ease of collection introduced by uncrewed (i.e., unmanned) aerial systems (UASs) have increased opportunities to gather image data outside of satellite re-visit cycles. The convenience and relatively low cost of many UAS platforms have made this method of remote sensing an attractive option for many researchers and agricultural service providers. Boursianis et al. [8] demonstrated the exponential growth in interest

for this technology over the preceding seven-year period based on Scopus results for keywords “UAS” and “smart farming.” However, a false perception has emerged in some sectors (particularly in new users who are now coming to remote sensing because of UAS) that flying under the clouds, as performed by UASs, produces cloud-free data. While technically true, in reality, when UASs operate in the presence of cloud shadows, variable lighting conditions reduce image quality and radiometric accuracy. These circumstances introduce other cascading effects into data products such as vegetation indices [9–11]. Although the response is index-specific, there is unequivocal evidence that many popular vegetation indices are sensitive to illumination differences [12,13]. Kaur [11] specifically investigated these effects for indices derived from UAS imagery. Other platforms, including satellite and handheld sensors, have been investigated as well. Researchers have also considered indices derived from both multispectral [11,12] and hyperspectral sensors [12]. This noted sensitivity has implications for users because these indices are often the basis for management decisions (Figure 1). Additionally, because many UAS image datasets are auto-rectified from manufacturer-provided in-field targets, or alternatively white-balanced by processing software, uneven illumination may affect the basic processing of imagery.



**Figure 1.** Effect of cloud cover on low-altitude UAS-collected five-band multispectral imagery (A) visible bands (blue, green, and red); (B) color infrared (green, red, near-infrared); (C) Normalized Difference Vegetation Index (green indicates more crop vigor), and (D) Modified Triangular Vegetation Index (green indicates more chlorophyll). The effect is clearly visible in both visible and near-infrared bands as dark pixels; hence, vegetation indices computed from these bands may be inaccurate. The severity of response varies with index but is especially pronounced in the estimation of chlorophyll shown in (D).

Kedzierski and Weirzbicki [14] proposed an index for the post-collection evaluation of UAS image quality. These authors highlighted that poor radiometric quality also hampered the generation of tie points, which are important in the pre-processing of UAS imagery, particularly when digital surface models are calculated. Because these products rely on matching features between individual image frames, cloud shadows moving across an

otherwise well-lit scene may be detrimental to the matching process. One drawback to their proposed method to rate the quality was the time involved in processing the image data. Intuitively, a practical solution to this limitation is to eliminate the need for such a process by avoiding image collection during periods where image quality is likely to be poor. This need may become more significant as agricultural technologies advance.

Currently, in the USA, federal restrictions dictate that UASs must be operated by human pilots. These pilots, potentially having no background in remote sensing or photogrammetry, make the decision about whether or not weather conditions are suitable for safe flight operations. These pilots thus make a de facto decision about the expected data quality. As the industry moves farther towards level 5 (i.e., fully) autonomous systems for farm management, the expectation is that machine systems will be remotely scheduled to collect data. While this scenario may presently seem out of reach, UAS companies are already envisioning this reality (e.g., American Robotics). For this reason, we propose that a system is needed which can serve as a gatekeeper for imaging by autonomous systems. In other words, we need a system check which reviews sky conditions and makes a determination if cloud presence will be a limiting factor in image quality. Coupled with an engineered solution, this technique could be used for remote scheduling of UASs missions.

Deep learning is a popular machine learning paradigm where the image feature vectors are learned through higher-level representations of image contents rather than engineered features such as shape, color, and texture. This method is primarily used to solve difficult pattern recognition tasks involving digital images and videos. Deep learning models take raw pixels of training images and learn useful features for further classification using artificial neural networks (ANN), which are sophisticated learning algorithms commonly used for complex image recognition tasks. The ANN is an interconnected network of computational processing nodes inspired by the way the human nervous system works. In this process, the digital imagery are learned by first transforming them into a hierarchical set of representations such as low, mid, and high-level features. These features are then learned using a fully connected ANN. However, image classification using ANN is limited by mainly two factors: (1) raw pixel data from training images do not provide a stable representation of objects in the images with respect to lighting, object size, object color, camera focus, camera angle, and other variations; and (2) the amount of computation (or neurons) required to learn such representations is enormously large and hence impractical.

Convolutional neural networks (CNN) are a type of deep learning ANN commonly used for image classification tasks. Researchers have applied CNN to computer vision challenges, developing methods that recognize traffic signs [15] and license plates [16]. Related to agriculture, CNN models have been used for tasks such as real-time apple detection in orchards [17]. With a detection accuracy of ~85%, the model was able to detect fruit in difficult scenarios such as the occlusion of a target, complex backgrounds, and indifferent exposures. A CNN transforms an input image into a feature map representation using a cascade of modules that perform three operations: (1) convolution filtering, (2) rectified linear unit (ReLU), and (3) pooling.

Deep learning architectures such as CNN could perform better by introducing more modules of these operations into the architecture. ResNet uses the same three operations as CNN. However, ResNet also utilizes skip connections and bypass loops to jump over multiple layers, mimicking pyramidal cells in the human cerebral cortex. In traditional deep learning networks such as CNN, the number of layers of image features is increased through convolutional filtering, and the resolution is decreased through pooling. ResNet constructs a deeper model by adding identity-mapping layers, while the other layers are copied from traditional (shallow) deep learning architecture. In this way, the deeper network constructed by this process will not produce a training error that is higher than the error rates of the shallower architecture. Thus, ResNet reduces the difficulty of training, resulting in smaller training and generalization errors [18,19]. Prior studies indicate that ResNet can produce better classification accuracy relative to other widely accepted methods [20,21].

Inexpensive small computers (e.g., NVIDIA Jetson Nano) now make it possible to run power-efficient artificial intelligence (AI) applications such as neural networks in an edge computing framework. “Edge computing” refers to a distributed computing framework that brings the application closer to the site where data are being acquired. Edge AI uses machine learning models to analyze data gathered by hardware in situ. In this approach, the computer processes data on-site rather than in the cloud. As a result, the latency and bandwidth needs are greatly reduced. This proximity of source data and powerful computing can be valuable in terms of real-time insights, improved response times, and better bandwidth availability. This has the effect of allowing real-time decision making by autonomous machines. A benchmark study demonstrated that these computers were capable of supporting real-time computer vision workloads for applications such as autonomous motor vehicle driving [22]. These small computers were further shown as beneficial in detecting drowsiness and inattention in motor vehicle drivers [23]. Edge computers have been paired with deep learning models and shown to produce suitable results. A class attendance system based on edge computing and deep learning produced efficient face localizations and detections within 0.2 s [24]. Such a pairing was also used to create MobileNet and PedNet, which, respectively, detected vehicles and pedestrians with an accuracy of approximately 85% [25]. A similar study validated the use of edge computing for the problem of construction vehicle detection using deep learning [26]. More relevant to UAS applications in agriculture, Ullah and Kim [27] demonstrated the capacity for these systems to handle the classification of 3D point-cloud and hyperspectral image data using deep learning models.

Following these observations, we had two objectives for this project. The first was to compare common deep learning approaches to classify sky conditions with regard to clouds in agricultural fields. The second was to develop an AI-based edge computing system to fully automate the classification process.

## 2. Materials and Methods

### 2.1. Data Collection

Data were collected using consumer-grade trail cameras (Strike Force HD Pro 18 MP, Browning Trail Cameras, Birmingham, AL, USA) installed at 13 sites across Mississippi (USA) in 2019 (Figure 2). Cameras were re-deployed at a subset of six sites for the 2020 growing season. Reasons for a smaller number of sites for 2020 are two-fold. First, a lack of differing weather conditions between sites was observed in 2019; thus, some sites were systematically removed due to their proximity to other sites. Second, as will likely be a common theme in much research coming from this period, travel restrictions related to the health crisis ultimately prevented researchers from traveling to some deployment sites.

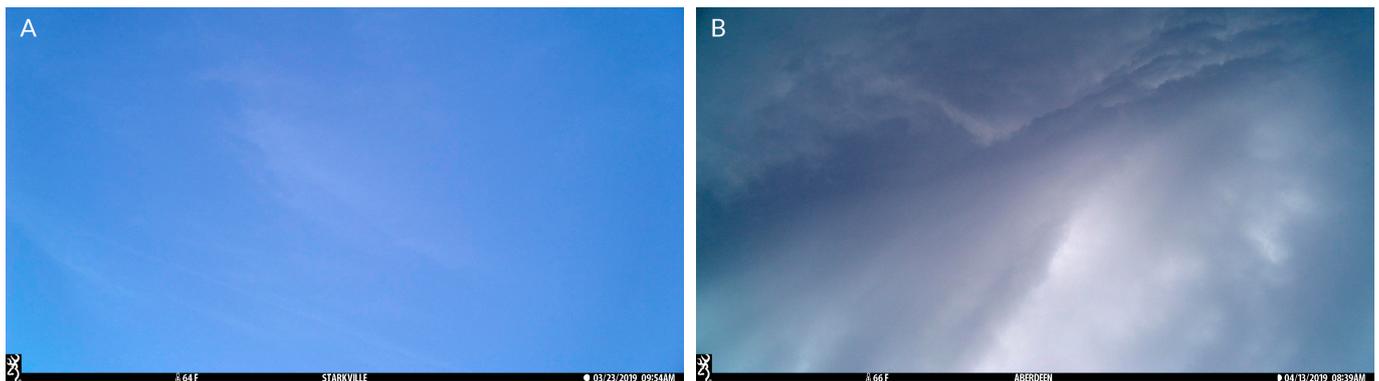
Cameras were angled to collect an oblique, unobstructed view of the sky (Figure 2). Cameras were placed in time-lapse mode and set to collect one image every hour on a daily basis. For the analysis, only images collected from 9 AM to 3 PM were utilized, and images outside this period were discarded. This hourly period was selected to represent the widest limit for UAS operation. With respect to sun angle and good remote sensing practices, this wide window would not be appropriate for all months of the year. This study was conducted from March through September to align with the crop growing season in the region. Thus, the window of collection was matched to the potential window for UAS use in agriculture for the geographic region of interest. From this collection, we utilized the complete image set, ~13,000 images, taken from the two-year period.



**Figure 2.** (A) The locations of trail cameras used to capture the sky conditions across Mississippi; (B) trail cameras were mounted to capture an oblique view of the sky at hourly intervals.

## 2.2. Data Labeling

All 13,000 images were labeled by a single human reviewer prior to network training. The overall task for the human reviewer was to view each image and indicate if the sky condition was favorable for high-quality image collection with a UAS. Each image was labeled according to the likelihood of (1) good image quality expected or (2) degraded image quality expected in the subsequent UAS operation (Figure 3). While on the surface, this may seem simplistic, the matter is more complicated than merely indicating if clouds are or are not in the field of view. The presence of clouds alone does not necessarily indicate reduced image quality. For instance, thin layers of largely transparent clouds may have no noticeable effect on image quality, while a solid cloud layer will attenuate the incoming light but will not result in uneven illumination in collected images. Thus, the decision is based on more than cloud presence but also altitude, density, and layering of clouds. The decision is also more than percent of cloud cover, as one poorly placed dense cloud could result in a dark streak across the mosaic. Therefore, any classifier must first decide if clouds are present within the image and then if the clouds present will be detrimental to overall image quality. For this reason, the labeling was performed by a subject matter expert with advanced degrees in operational meteorology, extensive background in remote sensing, and multiple years of experience operating UASs.



**Figure 3.** Example trail camera images showing (A) light cloud presence considered acceptable for acquiring aerial images using UAS; and (B) dense cloud presence considered unacceptable for acquiring aerial images using UAS due to the attenuated radiation and potential for uneven lighting of reflective surface.

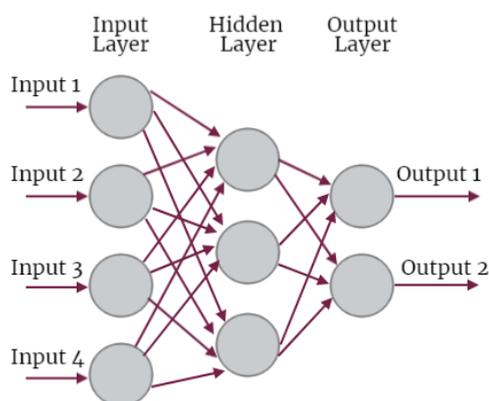
The labeling workflow was as follows. Using the available set of hourly trail camera images, each image was visually inspected for the presence of clouds. For the first level of classification, if any amount of cloud presence, however faint, was detected, the image was marked as having clouds. For the second level of classification, the subject matter expert rated each image determined to have clouds in the first classification for binary classification based on whether the sky condition was suitable for UAS collection of high-quality imagery.

### 2.3. Deep Learning Networks

As mentioned previously, ANN is commonly used for complex image recognition tasks through linear and nonlinear machine learning models. An ANN is comprised of an input layer, one or more hidden layers, and an output layer, as shown in Figure 4. The function of the input layer is to propagate its input to the next layer for processing. Each image used to train an ANN model is defined by image properties or features represented as a numerical feature vector ( $x$ ). A linear ANN learning model  $f(x)$  learns a transformation matrix ( $A$ ) using the feature vectors ( $x$ ) extracted from the training images and includes a bias term ( $b$ ). The model is represented by

$$f(x) = x_i A^T + b, \quad (1)$$

where the training image vector is represented as  $x_i$ ,  $i = \{1, 2, \dots, n\}$ . For each input feature vector, there is a weight vector  $a\{i\}$ , which indicates the strength of corresponding input to influence the output. An ANN is defined by the optimal set of weights or parameters learned through training process. To train the ANN, the weights are initialized with random values. The goal of the training process is to nudge the weight values in a direction that minimizes the deviation of the actual output of ANN from the desired output. Hidden layers perform the computation, as described in Equation (1). Circular blocks shown in the hidden layer(s) represent neurons as a computational unit. These units perform the calculations and propagate their output to the next hidden layer (if appropriate) or the output layer. This third and final layer in the ANN gives the desired output. To test the learned model, the ANN is presented with a new set of feature vectors to determine how well the network generalizes the data it learned during the training process.

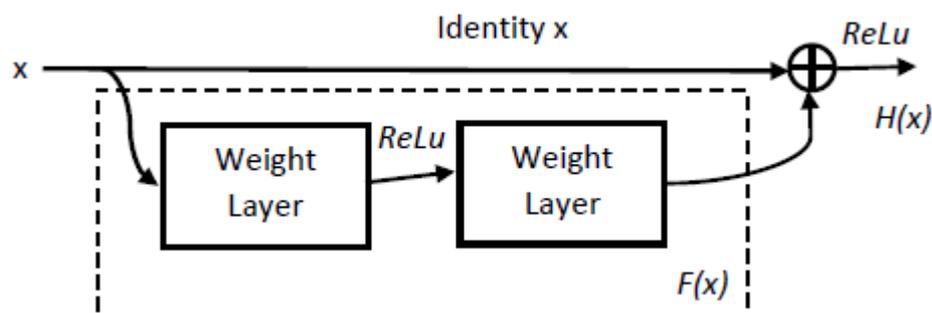


**Figure 4.** A generalized illustration of a simple three-layered feed-forward artificial neural network.

A CNN is a series of convolutional filters with learnable kernel values, followed by a fully connected ANN. These “kernels” are represented as matrices applied to the input images. The convolutional filters transform an input image into a feature map representation. During the training process, CNN learns the optimal values for kernel functions to enable the extraction of useful features from the input map. In each module, CNN could employ or learn more than one filter to efficiently extract the feature maps. After convolutional filtering, CNN applies ReLU to extracted feature maps to introduce nonlinearity into the learning. The ReLU step is always followed by the pooling step, where the CNN downsamples the feature map to reduce the size and thereby the computation in following stages. The last step in the CNN is a fully connected ANN to learn the feature maps extracted through convolutional filters. As previously mentioned, ResNet utilizes the same three basic steps but overcomes some noted limitations through key modifications such as skip connections and bypass loops. These modifications perform identity mapping, and their outputs are added to the outputs of the modules, as shown in Figure 5. For a desired mapping  $H(x)$ , it can be shown that

$$H(x) = F(x) + x, \quad (2)$$

where  $F(x)$  represents a stacked nonlinear layer fit to another function. This identity mapping does not add additional parameters or computational complexity.



**Figure 5.** A generalized illustration of the modifications present in a deep residual network building block using identity bypass.

#### 2.4. Network Training

A set of 100 images was used for training; 50 randomly selected images that were labeled as “good image quality expected,” and 50 randomly selected images that were labeled as “degraded image quality expected.” The experimental study evaluated the performance of five classifiers, CNN, and four permutations of ResNet classifiers. We used pre-trained and non-pre-trained versions of both ResNet18 and ResNet34. ResNet models

were pre-trained using the ImageNet dataset. To ensure there was no bias in the classifiers due to selection of training images, each architecture was re-trained and re-tested with an additional five sets of randomly selected training images. Prior to training, we first converted the trail camera images collected from their native resolution of  $4000 \times 3000$  (~12 megapixels) pixels to more appropriate resolutions of  $32 \times 32$  pixels (CNN) and  $112 \times 112$  pixels (ResNet). All three models used in this study were optimized using standard gradient descent with momentum/Nesterov [28]. The CNN and ResNet models were trained on an Intel X86-based desktop computer (Intel Xeon W-2225 CPU at 4.10 GHz and 32 GB of RAM). Both pre-trained and non-pre-trained deep learning classifiers were based on the Pytorch implementations running on Windows 10 operating system.

This study first evaluated a five-phase CNN (Table 1). We used max pooling for this phase as it is the most common in the literature [29]. In max pooling, the output map is generated by extracting a maximum value of the feature map from extracted tiles of a specified size and stride.

**Table 1.** Architecture of 5-phase convolutional neural network used in this study.

Phase	Phase Name	Output Size (pixels)	Phase Information	Processing
1	Convolution 2D	$28 \times 28$	$5 \times 5, 6$ , stride 1	Input size $32 \times 32$ , ReLu
2	Pooling	$14 \times 14$	$2 \times 2$ max pooling, stride 2	
3	Convolution 2D	$10 \times 10$	$5 \times 5, 16$ , stride 1	ReLu, stride 1
4	Pooling	$5 \times 5$	$2 \times 2$ max pooling, stride 2	$2 \times 2$ max pool, stride 1
5	Fully connected ANN	$1 \times 1$	Loss = Cross Entropy Momentum = 0.9	ReLu

The same training set was introduced to all four permutations of the ResNet classifier, as detailed in Table 2.

**Table 2.** Architecture of 7-phase deep residual learning network used in this study.

Phase	Phase Name	Output Size (pixels)	ResNet18	ResNet34	Processing
1	Convolution 2D	$112 \times 112$	$7 \times 7, 64$ , stride 2 $3 \times 3$ max pooling, stride 2		Input size $112 \times 112$ , ReLu
2	Pooling	$56 \times 56$			
3	Convolution 2D	$56 \times 56$	$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 64 \\ 3 \times 3 & 64 \end{bmatrix} \times 3$	ReLu
4	Convolution 2D	$28 \times 28$	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 128 \\ 3 \times 3 & 128 \end{bmatrix} \times 4$	ReLu
5	Convolution 2D	$14 \times 14$	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 256 \\ 3 \times 3 & 256 \end{bmatrix} \times 6$	
6	Convolution 2D	$7 \times 7$	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3 & 512 \\ 3 \times 3 & 512 \end{bmatrix} \times 3$	
7	Fully connected ANN	$1 \times 1$	Loss = Cross Entropy Momentum = 0.9		Average pool, SoftMax

### 2.5. Transition to Edge Computer Platform

The concept of edge AI involves computing data in real-time using an embedded system. In supervised learning, the training processes usually require more computational resources when compared to the testing process. Thus, the edge AI is typically used for the testing/inference step, and training is performed on a dedicated desktop workstation.

Then, the trained model with the obtained weights is deployed on the embedded hardware for real-time edge computing. In this study, we utilized an NVIDIA Jetson Nano Model P3450 with quad-core ARM A57 CPU at 1.43 GHz and 128-core Maxwell GPU with 4 GB of RAM and evaluated the capacity of such a system to perform both training and testing. We used the same image dataset and followed the same procedures outlined in Section 2.4 with this new hardware to conduct the experiment. We also tested and studied an 8-megapixel Raspberry Pi camera, directly interfaced with the Jetson Nano running the CNN, ResNet18, and ResNet34 classifiers. The Nano with the camera attached was placed outside, with the camera facing the sky in a way comparable to our trail cameras. Ten images were taken at one-second intervals and provided to each of the pre-trained classifiers. The results from these tests were displayed on the monitor detailing the real-time decision on if the sky conditions were likely to produce good image quality and were thus appropriate for UAS data collection.

### 2.6. Comparison to Human Model

We compared the performance of each deep learning model against the classification produced by the subject matter expert. Kappa statistic and overall accuracy were used as indicators of comparison. Kappa is a discrete multivariate technique commonly used in remote sensing studies to measure agreement between two or more classifiers [30]. Overall accuracy represents the proportion of samples that are classified correctly. In a confusion matrix, overall accuracy is the sum of the diagonal elements divided by the total sum of the matrix. The Kappa ranges from zero (representing random chance) to one (perfect agreement).

## 3. Results

The overall accuracy of the CNN classification for learning rates of 0.01, 0.001, and 0.0001 was 89.31%, 89.95%, and 82.64%, respectively. The corresponding Kappa statistics were 0.64, 0.67, and 0.52, respectively. Based on these Kappa statistics, there is moderate (<0.6) to substantial (>0.6) agreement between our classifiers. To further validate these findings, other permutations were investigated. Small changes to feature sizes did not produce significant differences in accuracies. The deep learning classifiers were also studied with different training set sizes. Our results indicated no significant improvement in accuracy beyond 50 training images per class. The insignificant variance between training sets validated that similar accuracy was obtained regardless of the training set used, thereby confirming no bias.

The number of epochs for these experiments was determined empirically. In both cases, the CNN classifier required ~30 epochs to reach the near-final accuracies (Figure 6). Experiments with other learning rates, number of epochs, and layers did not produce a better overall accuracy and Kappa statistic. We also investigated these classifiers with a much higher number of epochs. We found that the overall accuracy and Kappa statistic showed little improvement when the number of epochs exceeded 50 for CNN classifiers.

ResNet classifiers exhibited considerably higher overall accuracy within fewer epochs when compared to CNN. Among two different ResNet configurations, the 34-layered network produced the best classification accuracy and Kappa statistic. The overall accuracy of the classification for a learning rate of 0.001 for ResNet18 and ResNet34 was 91.7% and 92%, respectively, with corresponding Kappa statistics of 0.75 and 0.77. Based on these Kappa statistics, there is substantial agreement between our classifiers. In all the cases, the ResNet classifiers required ~15 epochs to reach the near-final accuracies (Figure 7). No significant improvement was seen beyond 15 epochs. Experiments with other learning rates and numbers of epochs did not produce a better overall accuracy and Kappa statistic.

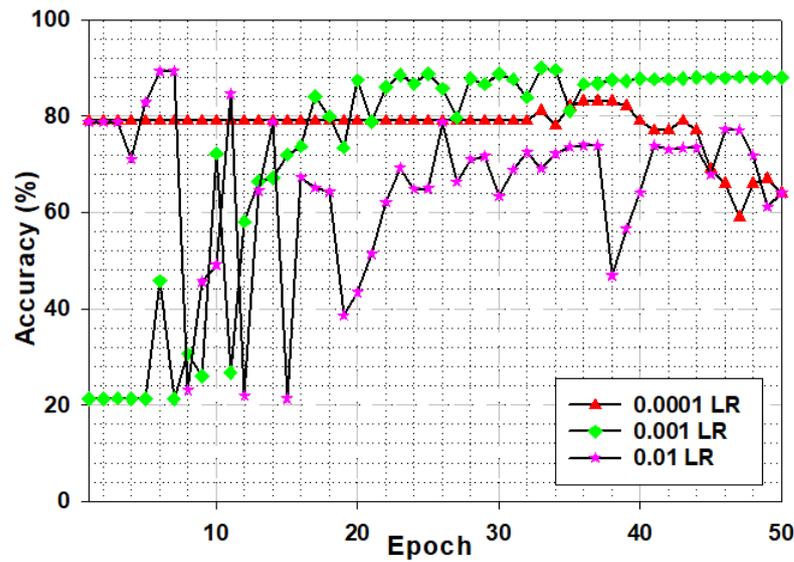


Figure 6. Accuracy by epoch for three learning rates (LR) used for training the convolutional neural network.

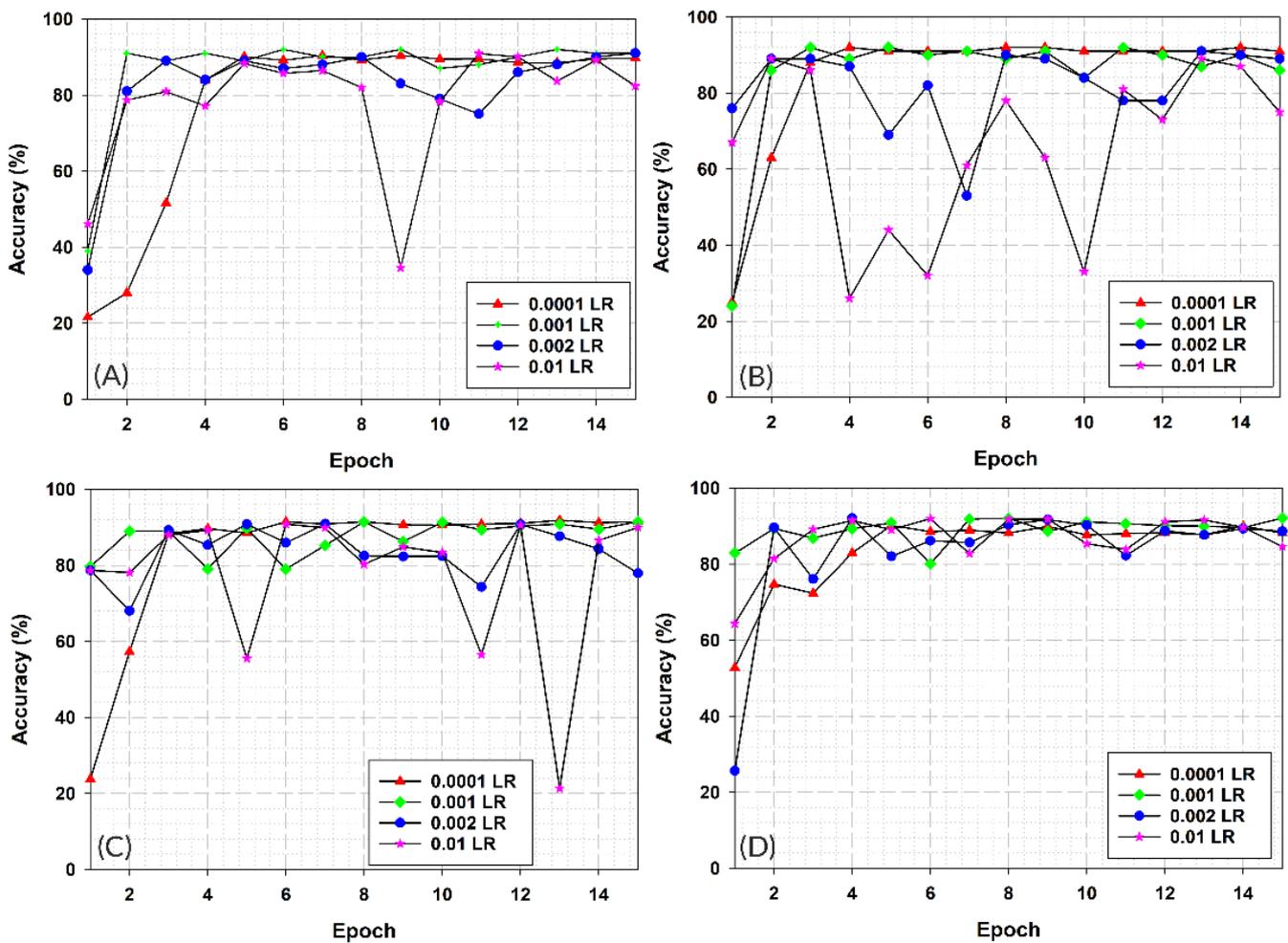


Figure 7. Training on deep residual networks with 18 layers using four different learning rates (A) non-pre-trained ResNet-18; (B) pre-trained ResNet-18; (C) non-pre-trained ResNet-34; (D) pre-trained ResNet-34.

The overall accuracy and Kappa statistic numbers produced by the AI edge computing framework were identical to those produced by the desktop platform. Moreover, we were able to demonstrate that the NVIDIA Jetson Nano edge computer could capably handle testing phases. Processing times varied according to the neural network (Table 3).

**Table 3.** Computational times to convergence for neural networks tested by platform.

Method	Kappa Statistic	Overall Accuracy (%)	Learning Rate	Training Time on Desktop Computer (min)
ResNet 18–Pre-trained	0.75	91	0.002	345
ResNet 18–Non-pre-trained	0.72	91	0.0001	274
ResNet 34–Pre-trained	0.75	92	0.001	454
ResNet 34–Non-pre-trained	0.77	92	0.001	279
CNN	0.67	90	0.001	855

#### 4. Discussion

The proposed approach was found to be more than capable of producing an accurate edge computing solution to sky classification. The approach was robust to the training image set in terms of both the specific images selected and the number of images. The CNN classifier, although capable of learning, was not as robust for this challenge as ResNet alternatives. Previous research suggests that we cannot improve the overall accuracy of CNN with regard to this image training set by altering parameters such as epochs, learning rate, and the number of layers [31]. This is due to the fact that plain deep networks such as CNN can have exponentially low convergence rates and thus are very difficult to optimize beyond a certain limit [31]. The complexity of the problem is also a factor that affects the convergence. In this study, we challenged the classifiers to separate images where clouds contain fuzzy samples that the CNN may not learn efficiently. To explore a more complex learning method and use deeper features, we recommend deeper learning classifiers such as ResNet. Our experiments demonstrated the capability of ResNet to produce higher accuracies, likely due to the identity connections solving vanishing gradients during the training process. The accelerated training of ResNet classifiers is evidenced by the lower number of epochs taken to achieve the best accuracy. The differences between pre-trained and non-pre-trained ResNet models, however, were minimal. We believe this is due to the fact that the ImageNet dataset used for pre-training does not contain images involving sky conditions.

In comparison to the human classifier, the deep learning approach has significant time advantages. Encouragingly, the ResNet classifiers were not only accurate representations of the subject matter expert’s classification, but they were also fully capable of being implemented in a low-cost edge computing framework. They were also capable of providing results in a reasonable time. The AI classifiers far surpassed the speed of human classification in all tests, indicating obvious efficiency gains. The pairing of the Jetson nano with the Raspberry Pi camera demonstrates that this system can produce results in real-time. Such a model is a meaningful advance in the overall progress towards fully automated technologies for the farming environment. The application of information and communications technologies (e.g., internet of things, big data analytics, machine learning) to agricultural interests will continue to increase the potential for UASs to make meaningful contributions to the field; a trend that has been referred to as Farming (or Agriculture) 4.0 [8].

Taking a larger view, there are some potential pitfalls that should be noted. Currently, fallible humans decide when the time is right for flight. There are advantages and disadvantages to this model. With a human operator, it may be apparent to the experienced UAS pilot that a hole in the clouds will open up shortly, just long enough to gather the needed information. The gatekeeping system we envision herein might not have the ability to capitalize on such an opportunity. However, the current reality is that only researchers have the luxury of waiting for the said hole, but few service providers will have such

freedom. In order to meet the demands on their time, many have limited time in the field to be patient. Therefore, this potential shortcoming may not be universally relevant or significant in the grander scheme.

It should also be acknowledged that the quality standard and investment in obtaining high-quality data is a different proposition between researchers and service providers and also variable for different agronomic problems. Some applications have little need for high-quality, evenly illuminated imagery (e.g., visually inspecting crop loss following a hail storm). Humans may natively intuit this extraneous information, while autonomous UASs will need programming to be responsive to the level of quality dictated by the application. Externalities such as these would need to be accounted for in a more final product. This suggests more AI development will be needed to support decision-making processes that account for the context of the image collection. As the dependence of agriculture on autonomy increases, so too will the need for the collection of high-quality data as an input for machine learning systems tasked with farm management and informing decision-making practices that improve the profitability and sustainability of production agriculture.

Finally, it is important to stress that this method is intended to prevent data collection through the identification of poor sky conditions. It is not a method for cloud removal from UAS imagery. Additional research paths are needed to identify solutions that preserve information in collected imagery arising from those where obvious quality issues exist.

## 5. Conclusions

Cloud shadows are a significant determinant in the overall quality of imagery that are collected by UASs. The myriad of advances in co-working between humans and robots, and the integration of fleets of robots, suggest that increasingly, many farm management tasks will be delegated partially, if not fully, to autonomous machines. When the human factor is removed, more quality control elements will be warranted to protect data quality, especially in situations where these data will dictate crop management actions. Our experiments demonstrated that such a system can be developed with existing hardware and deep learning methods. The low-cost system we have demonstrated was most powerful when ResNet classifiers were imported to AI-based edge computing frameworks. The methodology described in this research is not limited to the classification of sky conditions but could also be implemented for other applications where the detection of atmospheric variables in real-time is needed.

**Author Contributions:** Conceptualization, J.M.P.C., S.S. and L.L.W.; methodology, J.M.P.C. and S.S.; software, M.Z. and C.D.M.; validation, M.Z. and C.D.M.; formal analysis, M.Z.; investigation, J.M.P.C. and L.L.W.; resources, J.M.P.C.; data curation, J.M.P.C., L.L.W. and M.Z.; writing—original draft preparation, J.M.P.C. and S.S.; writing—review and editing, M.Z., C.D.M. and L.L.W.; visualization, J.M.P.C. and S.S.; supervision, J.M.P.C. and S.S.; project administration, J.M.P.C. and S.S.; funding acquisition, J.M.P.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This publication is a contribution of the Mississippi Agricultural and Forestry Experiment Station. This material is based upon work that is supported by the National Institute of Food and Agriculture, US Department of Agriculture, Hatch project under accession number 721150.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are openly available in the Mississippi State University Libraries Institutional Repository as Czarnecki et al. (2021). Mississippi Sky Conditions. Available at <https://scholarsjunction.msstate.edu/cals-publications/27> (accessed on 19 September 2021).

**Acknowledgments:** The authors wish to thank trail camera hosts for access to their property. Additional appreciation is extended to Katie Clarke for assistance with the graphical abstract.

**Conflicts of Interest:** The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Thorp, K.; Tian, L. A Review on Remote Sensing of Weeds in Agriculture. *Precis. Agric.* **2004**, *5*, 477–508. [[CrossRef](#)]
2. Dadhwal, V.; Ray, S. Crop Assessment Using Remote Sensing-Part II: Crop Condition and Yield Assessment. *Indian J. Agric. Econ.* **2000**, *55*, 55–67.
3. Ozdogan, M.; Yang, Y.; Allez, G.; Cervantes, C. Remote Sensing of Irrigated Agriculture: Opportunities and Challenges. *Remote Sens.* **2010**, *2*, 2274–2304. [[CrossRef](#)]
4. Kasampalis, D.A.; Alexandridis, T.K.; Deva, C.; Challinor, A.; Moshou, D.; Zalidis, G. Contribution of Remote Sensing on Crop Models: A Review. *J. Imaging* **2018**, *4*, 52. [[CrossRef](#)]
5. Chauhan, S.; Darvishzadeh, R.; Boschetti, M.; Pepe, M.; Nelson, A. Remote Sensing-Based Crop Lodging Assessment: Current Status and Perspectives. *ISPRS J. Photogramm.* **2019**, *151*, 124–140. [[CrossRef](#)]
6. Eberhardt, I.D.R.; Schultz, B.; Rizzi, R.; Sanches, I.D.; Formaggio, A.R.; Atzberger, C.; Mello, M.P.; Immitzer, M.; Trabaquini, K.; Foschiera, W. Cloud Cover Assessment for Operational Crop Monitoring Systems in Tropical Areas. *Remote Sens.* **2016**, *8*, 219. [[CrossRef](#)]
7. Ju, J.; Roy, D.P. The Availability of Cloud-Free Landsat ETM+ Data over the Conterminous United States and Globally. *Remote Sens. Environ.* **2008**, *112*, 1196–1211. [[CrossRef](#)]
8. Boursianis, A.D.; Papadopoulou, M.S.; Diamantoulakis, P.; Liopa-Tsakalidi, A.; Barouchas, P.; Salahas, G.; Karagiannidis, G.; Wan, S.; Goudos, S.K. Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in Smart Farming: A Comprehensive Review. *Internet Things* **2020**, 100187. [[CrossRef](#)]
9. Wierzbicki, D.; Fryskowska, A.; Kedzierski, M.; Wojtkowska, M.; Delis, P. Method of Radiometric Quality Assessment of NIR Images Acquired with a Custom Sensor Mounted on an Unmanned Aerial Vehicle. *J. Appl. Remote Sens.* **2018**, *12*, 015008. [[CrossRef](#)]
10. Kedzierski, M.; Wierzbicki, D.; Sekrecka, A.; Fryskowska, A.; Walczykowski, P.; Siewert, J. Influence of Lower Atmosphere on the Radiometric Quality of Unmanned Aerial Vehicle Imagery. *Remote Sens.* **2019**, *11*, 1214. [[CrossRef](#)]
11. Kaur, S. Handling Shadow Effects on Greenness Indices from Multispectral UAV Imagery. Master's Thesis, School of Forest Science and Resource Management, Technical University of Munich, Munich, Germany, 2020.
12. Zhang, L.; Sun, X.; Wu, T.; Zhang, H. An Analysis of Shadow Effects on Spectral Vegetation Indexes Using a Ground-Based Imaging Spectrometer. *IEEE Geosci. Remote Sens.* **2015**, *12*, 2188–2192. [[CrossRef](#)]
13. Martín-Ortega, P.; García-Montero, L.G.; Sibelet, N. Temporal Patterns in Illumination Conditions and Its Effect on Vegetation Indices Using Landsat on Google Earth Engine. *Remote Sens.* **2020**, *12*, 211. [[CrossRef](#)]
14. Kedzierski, M.; Wierzbicki, D. Radiometric Quality Assessment of Images Acquired by UAV's in Various Lighting and Weather Conditions. *Measurement* **2015**, *76*, 156–169. [[CrossRef](#)]
15. Han, Y.; Oruklu, E. Traffic Sign Recognition Based on the Nvidia Jetson Tx1 Embedded System Using Convolutional Neural Networks. In Proceedings of the 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Medford, MA, USA, 6 August 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 184–187.
16. Lee, H.; Grosse, R.; Ranganath, R.; Ng, A.Y. Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14 June 2009; ACM: New York, NY, USA, 2009; pp. 609–616.
17. Mazzia, V.; Khaliq, A.; Salvetti, F.; Chiaberge, M. Real-Time Apple Detection System Using Embedded Systems with Hardware Accelerators: An Edge AI Application. *IEEE Access* **2020**, *8*, 9102–9114. [[CrossRef](#)]
18. Allen-Zhu, Z.; Li, Y. Backward Feature Correction: How Deep Learning Performs Deep Learning. *arXiv Prepr.* **2020**, arXiv:2001.04413.
19. Li, S.; Jiao, J.; Han, Y.; Weissman, T. Demystifying Resnet. *arXiv Prepr.* **2016**, arXiv:1611.01186.
20. Jung, H.; Choi, M.-K.; Jung, J.; Lee, J.-H.; Kwon, S.; Young Jung, W. ResNet-Based Vehicle Classification and Localization in Traffic Surveillance Systems. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 61–67.
21. Liu, S.; Tian, G.; Xu, Y. A Novel Scene Classification Model Combining ResNet Based Transfer Learning and Data Augmentation with a Filter. *Neurocomputing* **2019**, *338*, 191–206. [[CrossRef](#)]
22. Otterness, N.; Yang, M.; Rust, S.; Park, E.; Anderson, J.H.; Smith, F.D.; Berg, A.; Wang, S. An Evaluation of the NVIDIA TX1 for Supporting Real-Time Computer-Vision Workloads. In Proceedings of the 2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), Pittsburgh, PA, USA, 18 April 2017; pp. 353–364.
23. Reddy, B.; Kim, Y.-H.; Yun, S.; Seo, C.; Jang, J. Real-Time Driver Drowsiness Detection for Embedded System Using Model Compression of Deep Neural Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21 July 2017; pp. 121–128.

24. Pomsar, L.; Kajati, E.; Zolotova, I. Deep Learning Powered Class Attendance System Based on Edge Computing. In Proceedings of the 18th International Conference on Emerging eLearning Technologies and Applications (ICETA), Košice, Slovenia, 12 November 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 538–543.
25. Barba-Guaman, L.; Eugenio Naranjo, J.; Ortiz, A. Deep Learning Framework for Vehicle and Pedestrian Detection in Rural Roads on an Embedded GPU. *Electronics* **2020**, *9*, 589. [[CrossRef](#)]
26. Arabi, S.; Haghighat, A.; Sharma, A. A Deep-Learning-Based Computer Vision Solution for Construction Vehicle Detection. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 753–767. [[CrossRef](#)]
27. Ullah, S.; Kim, D. Benchmarking Jetson Platform for 3D Point-Cloud and Hyper-Spectral Image Classification. In Proceedings of the 7th IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Korea, 19 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 477–482.
28. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the Importance of Initialization and Momentum in Deep Learning. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 17 June 2013; Microtome Publishing: Brookline, MA, USA, 2013; pp. 1139–1147.
29. Sun, M.; Song, Z.; Jiang, X.; Pan, J.; Pang, Y. Learning Pooling for Convolutional Neural Network. *Neurocomputing* **2017**, *224*, 96–104. [[CrossRef](#)]
30. Viera, A.J.; Garrett, J.M. Understanding Interobserver Agreement: The Kappa Statistic. *Fam. Med.* **2005**, *37*, 360–363. [[PubMed](#)]
31. 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 Workshops, Las Vegas, NV, USA, 27 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 770–778.