



# Detection of tartrazine colored rice flour adulteration in turmeric from multi-spectral images on smartphone using convolutional neural network deployed on PaaS cloud

Madhusudan G. Lanjewar<sup>1</sup> · Pranay P. Morajkar<sup>2</sup> · Jivan Parab<sup>1</sup>

Received: 21 September 2021 / Revised: 9 December 2021 / Accepted: 25 January 2022 /

Published online: 3 March 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

Food adulteration occurs globally, in many facets, and affects almost all food commodities. Adulteration is not just a crucial economic problem, but it may also lead to serious health problems for consumers. Turmeric (*Curcuma longa*) is a world-class spice commonly contaminated with various chemicals and colors. It has also been used extensively in many Asian curries, sauces, and medications. Different traditional approaches, such as chemical and physical methods, are available for detecting adulterants in turmeric. These approaches are rather time-consuming and inaccurate methods. Therefore, it is of utmost importance to identify the adulterants in turmeric accurately and instantly. A cloud-based system was developed to detect adulteration in adulterated turmeric. The dataset consists of spectral images of turmeric with tartrazine-colored rice flour adulterant. Adulterants in weight percentages of 0%, 5%, 10%, and 15% were mixed with turmeric. A convolutional neural network (CNN) was implemented to detect adulteration, which achieved 100% accuracy for training and 94.35% accuracy for validation. The deep CNN (DCNN) models, namely, VGG16, DenseNet201, and MobileNet, were implemented to detect adulteration. The proposed CNN model outperforms DCNN models in terms of accuracy and layers. The CNN model is deployed to the platform as a service (PaaS) cloud. The deployed model link can be accessed using a smartphone. Uploading the adulterated turmeric image to a cloud link can analyze and detect adulteration.

---

✉ Jivan Parab  
jsparab@unigoa.ac.in

Madhusudan G. Lanjewar  
madhusudan@unigoa.ac.in

Pranay P. Morajkar  
pranay@unigoa.ac.in

<sup>1</sup> School of Physical and Applied Sciences, Goa University, Taleigao Plateau, Goa 403206, India

<sup>2</sup> School of Chemical Sciences, Goa University, Taleigao Plateau, 403206 Goa, India

**Keywords** Adulteration · Tartrazine · Turmeric · CNN · Cloud computing · PaaS cloud

## 1 Introduction

Turmeric is also called *Curcuma longa* and belongs to Asia and India in particular. Since ancient times, tuberous rhizomes and underground turmeric have been used as condiments, colors, and aromatic stimulants in several medicines [55]. Turmeric is an important spice produced in India and consumes nearly 80% of the world's population [55]. India is the world's leading turmeric producer and exporter. Turmeric areas cover approximately 6% of India's total spice and condiment area. Turmeric is also grown in China, Myanmar, Bangladesh, and Nigeria [55]. Turmeric and curcumin have been used traditionally to treat several ailments, such as acne, injuries, ambushes, eye infections, skin conditions, stress, and depression [22]. The Hindu literature called "Ayurveda" has spoken for thousands of years about this spice because of its ability to fight sickness and inflammatory conditions [7]. It has been extensively utilized in Ayurvedic medicines due to its bioactive properties, such as antioxidant, analgesic, anti-inflammatory, and antiseptic properties [30]. Thus, because of its popularity, there is increased demand for turmeric in world trade. Unfortunately, the quality of turmeric powder has been affected due to commonly used adulterants for undue profit.

Food quality is reduced by mixing added components that are either of low nutritional quality or could even be hazardous to human health. In turmeric and other spice powders, azo compounds such as Metanil yellow, Tartrazine, and Sudan dyes can be mixed to enhance turmeric colors [11, 12]. Azo dye is an organic compound with azo linkage and extended conjugation that facilitates light absorption in the visible region. Tartrazine, also called lemon yellow, is studied due to its severe harmful effects, such as inducing asthma and chronic hives. The acceptable daily intake (ADI) for tartrazine is estimated to be approximately 7.5 mg/kg/day by JECFA (1964). Moreover, several countries have banned its use in recent times [67]. A major problem with several of these azo dyes is that they produce aromatic amines and amino phenols in the gastrointestinal tracts during metabolism and hence are regarded/suspected as carcinogens based on both in vivo and in vitro studies on mice. A detailed account of the toxicity of azo dyes as a pollutant and their remediation strategies are very well explained in the works of Morajkar et al., and readers are directed to the following references [46, 48, 49] for further reading. Thus, it can be unambiguously concluded that adulteration of dyes such as tartrazine in turmeric is potentially life-threatening, and therefore, instant and accurate detection of such adulterants at an early stage is extremely essential.

Various conventional methods, such as rapid color testing [60], thin-layer chromatography, micellar chromatography, and high-performance liquid chromatography for the detection of adulterants in turmeric, have been reported [3, 19, 60]. Several analytic methods, such as higher-level liquid chromatography, polymerase chain reaction, high-performance capillary electrophoresis, and HPC ionizing mass spectrometry [10, 65, 72], have been applied to detect additives and chemical contaminants in turmeric. These methods require skilled operators and fresh samples in techniques such as HPLC. However, these traditional methods are highly accurate and offer satisfactory detection limits. The field usability of these methods is restricted due to their operational complexity, sample destructive methodologies, chemical requirements, postprocessing complexity, and difficulty in automating the detection process. Therefore, hyperspectral and multispectral imaging techniques are used for food adulterant detection. Various multispectral imaging techniques have been reported with fewer spectral bands that

improve the cost efficacy and reduce the computational burden. Imaging systems are helpful to detect food adulterants such as mincemeat adulterated with horsemeat [56], sugar adulteration in tomato paste [42], fraudulent replacement of hawk beef [15], and some other foods [31, 43, 71]. Not only image features but also semantic concepts are required for accurate hyperspectral image categorization [59]. These spectral band images are processed with image processing algorithms based on principal component analysis (PCA), linear discriminant analysis, Fisher's discriminant analysis (FDA), and the support vector machine (SVM) [57]. Several comparative analyses of ML models and performance indicators were discussed by Fadda et al. [17, 18]. Various modified efficient optimization algorithms for denoising the images were proposed by Kumar et al. [36, 37]. Cloud computing is a modern technology that has a huge impact on our lives. Cloud computing has a number of advantages, including the ability to quickly create centralized knowledge bases, allowing various embedded systems (such as sensor or control devices) to share intelligence [44]. Furthermore, complicated operations may be performed on low-spec devices simply by offloading processing to the cloud, which has the added benefit of saving energy. Computer services and facilities can be accessed anytime and anywhere using this technology. The United Nations Food and Agriculture Organization (FAO) introduced studies on Artificial Intelligence (AI) and ML in the entire agriculture sector [45]. AI and ML in cloud base systems are useful agriculture [45] and remote sensing [50].

In AI, ML algorithms are used to understand relationships between data and make judgments without explicit knowledge [41]. DL is a challenge for Internet of Things (IoT) hardware with restricted computing capability [39]. Cloud computing is one of the resources considered in the image processing area to process remotely sensed images [40]. Cloud computing tools are now available from cloud service vendors such as Amazon Web Services (AWS), Google's Compute Engine, Microsoft Azure, and Heroku for users on a "pay as you go" model. These tools can be used to perform tasks related to image processing, such as PaaS (platform as a service), SaaS (software as a service), or IaaS (infrastructure as a service) [2]. Many works related to remote sensing fields using satellite imaging can be found in the literature. In the remote sensing area, a cloud computing platform such as Google Earth Engine (GEE) was employed in addition to artificial neural networks (ANNs) [2]. Therefore, the use of cloud computing platforms for instant analysis and detection of adulterants in spices could prove to be very economical and beneficial to consumers.

## 1.1 Related work

Tartrazine (3-carboxy-5-hydroxy-1-(4'-sulfophenyl)-4-((4'-sulfophenyl) azo) pyrazole trisodium salt) is an artificial azo colorant that can be found in food items such as dairy products, chocolates, bread products, and beverages. Tartrazine (TZ) has been shown to have harmful health consequences when ingested in excess. Therefore, monitoring the TZ content in food items is essential.

Zoughi et al. proposed an assay for the selective analysis of tartrazine [74]. Fluorophores, carbon dots, and tartrazine were placed in a molecularly imprinted polymer matrix to produce an optical nanosensor. A variety of methods were used to characterize the synthesized carbon dots contained in the molecularly imprinted polymer. Saffron ice cream and fake saffron were successfully detected using a developed nanosensor. The nanosensor and HPLC analysis results were almost the same with a 95% confidence level. For the measurement of tartrazine and sunset yellow in food samples, Sha et al. suggested a simple and successful approach, linking ionic liquid-based aqueous two-phase systems with HPLC [58]. For both analytes, IL-

ATPSs had an extraction efficiency of 99% under optimal conditions. Yang et al. investigated fluorescence resonance energy transfer (FRET) between tartrazine and 3-mercapto-1,2,4-triazole-capped gold nanoclusters (TRO-AuNCs) to determine the TZ in foodstuff samples [70]. The proposed method was successfully applied to the determination of TZ in juice and honey samples with an excellent extraction efficiency of  $\sim 92.0\%$ , and precision (*Relative Standard Deviation (RSD)* 1.14~2.84%) was attained. These methods are time-consuming and require many sample modifications. They also have poor sensitivity and selectivity. The development of affordable diagnostic instruments for the measurement of this analyte is therefore urgently needed.

The use of quick, reliable, and noncontact technology products in the food industry has attracted considerable attention due to an increasing tendency toward online monitoring of food quality, safety standards, and authenticity. Hyperspectral imagery (HSI) is a nondestructive technique based on spatially resolved spectroscopy. Hyphenation of HSI with other measuring instruments could be a promising strategy to collect different types of information, such as the appearance, nature, microstructure, and special features of food products and even adulterants [33]. Bertelli et al. proposed diffuse reflectance mid-infrared Fourier transform spectroscopy (DRIFTS) to classify 82 honey samples [6]. Furthermore, Parvathy et al. employed a DNA barcoding technique to detect herbal adulterants in turmeric powder [53]. Although the DNA barcoding technique can identify some additives, this method is purely qualitative and requires a sequence of potential biological adulterants. Di Anibal et al. [13] employed high-resolution  $^1\text{H}$  nuclear magnetic resonance spectroscopy (NMR) to detect five types of Sudan dyes in various commercially available species, including turmeric. The basic requirement of the NMR technique is that the sample should be of very high purity and should be soluble in solvents such as dimethyl sulfoxide (DMSO) or  $\text{CHCl}_3$ , along with complex spectral processing to detect adulterant products. This method is not reliable for the accurate identification or quantification of adulterants. The CNN-based method was proposed by Izquierdo et al. [29] to detect adulteration of extra virgin olive oil (EVOO). Thermal graphic images of EVOO with various adulterants, such as olive pomace oil, refined olive oil, and sunflower oil, were employed for the analysis and classification. Eight CNN models were developed for the classifications and adulterant concentration estimation in EVOO and achieved accuracies ranging from 97 to 100%. Here, they have not given the size of the model. Therefore, the presence of adulterants in various food and food products, such as meat, honey, sauce, and spices, can be detected by developing hyperspectral or multispectral imaging systems with image processing algorithms. Chaminda Bandara et al. proposed a multispectral imaging system to identify the 21 common adulterants in turmeric [5]. They detected pure and adulterated samples only, not adulteration levels. Prabhath et al. [54] developed multispectral imaging and multivariate techniques to identify adulteration levels in turmeric powder. The adulteration level was predicted using the calibrated model with  $R^2=0.9644$  for every validation sample. DRIFTS combined with chemometrics was proposed by Hu et al. [27] to check 1200 black pepper samples adulterated by sorghum and Sichuan pepper, with doped levels from 5 to 50% combined with the pure sample. PCA, genetic algorithm optimized support vector machine (GA-SVM), and partial least squares discriminant analysis (PLS-DA) were used to treat the collected data. GA-SVM and PLS-DA models predict 100% correct results on the training and validation set of a pure black pepper sample. The total detection rates for the GA-SVM and PLS-DA models for the fake black pepper sample were more than 98% for the training set and more than 96% for the validation dataset. They employed GA-SVM and PLS-DA models to detect adulteration and mislabeling of

pepper only. For saffron adulteration detection, Kiani et al. [32] proposed a computer vision system (CVS) and an electronic nose (e-nose). They demonstrated that the proposed system can distinguish between authentic and adulterated samples only.

From the above literature, it is clear that different types of ML algorithms were developed to detect adulterants in turmeric. The majority of investigations uses spectrophotometer and image preprocessing with an ML model for adulterant identification in turmeric. A limitation of previous works is that these methods require a skilled person to handle. The researchers proposed ML-based methods and achieved acceptable accuracy, but the model size was not given, which is very important in deployment to the cloud or embedded devices. Therefore, there is scope for developing a scheme that would accurately and instantly detect adulterants. The motivation is primarily to design a software solution that might help companies and users to identify adulterants in turmeric. Python computer language was used to implement the CNN and DCNN models. The main contribution of this work includes:

- Proposing an efficient structure by selecting parameters and hyperparameters for a CNN network that is suitable for adulterant classification.
- Implementation of DCNN models with the same parameters and hyperparameters, which are used in CNN.
- Evaluating the proposed CNN with Adam and RMSprop optimizers using a confusion matrix. Comparing these models with the performance of DCNN models such as VGG16, DenseNet201, and MobileNet.
- Different combinations of optimizers were tried, and the most effective model was proposed.
- The modified CNN model has been used to detect adulteration in turmeric with an accuracy of 100% for training and 94.35% for validation, which is quite good compared to the literature reports.
- It may also be noted that none of the researchers have reported the size of the model, which is very important to deploy the model on to the embedded platform and cloud.
- Therefore, the proposed CNN model has been designed of a smaller size with more than 98% fewer parameters compared to other models. This facilitates deployment of the model on embedded platforms such as Raspberry pi and Jetson Nano in a much more effective manner.

## 2 Materials and methods

The framework of the proposed system to detect rice flour adulteration in turmeric using CNN is shown in Fig. 1.

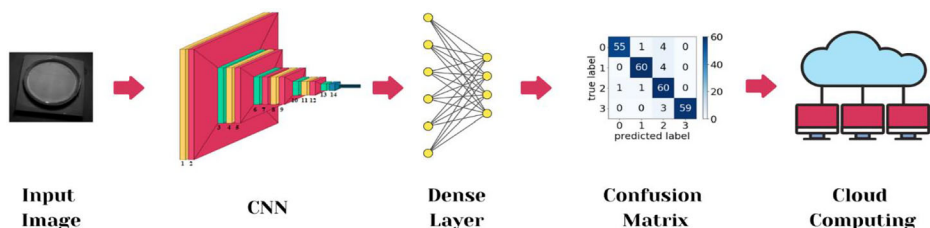


Fig. 1 Cloud-based framework to detect tartrazine-colored rice flour adulteration in turmeric

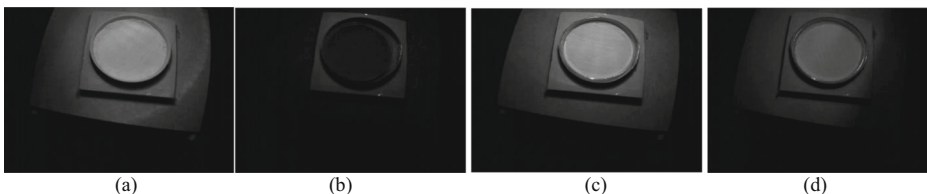
The dataset of multispectral images of adulterated turmeric is used in this work. These spectral images resized to  $224 \times 224$ . The dataset is split into two sections, the training and validation datasets. All the models are trained with a training dataset and validated with a validation dataset. The CNN and DCNN models were implemented using Python programming language. The confusion matrix technique was employed to evaluate the model performance. An efficient model is deployed to the cloud to classify mobile images.

## 2.1 Dataset details

The dataset consists of multispectral images of tartrazine-rice flour-adulterated turmeric powder samples [4]. The dataset contains multispectral images of thirty (30) replicates of authentic turmeric powder adulterated with tartrazine-colored rice flour. Adulterant was added to turmeric powder at concentrations of 0%, 5%, 10%, and 15% (w/w). There were four folders, namely, PER0, PER5, PER10, and PER15, which indicated 0%, 5%, 10%, and 15%, respectively, rice flour mixed with turmeric. Figure 2 shows the multispectral images of turmeric with rice flour adulterants in the dataset.

## 2.2 Multispectral imaging

Multispectral imagery is an expansion of white illumination to a better spectral resolution and delivers both spectral and spatial picture information. A multispectral imaging system with amplified effects is helpful to improve sensitivity, control, and calibrate the image intensity automatically. More spectral properties were provided by the multispectral camera with a specified optical filter. Multispectral images can act as the basis for the development of hyperspectral imaging that captures images in the electromagnetic spectrum at different wavebands and combines spectral signatures with the chemical compounds that produce them through the absorption of light frequencies that resonate in chemical connections. Hyperspectral (HS) and multispectral (MS) images have more bands (sometimes more than 10) than RGB images [38]. At least partly outside of the visible spectral range, the spectral regions cover portions of infrared and ultraviolet regions. For example, a multispectral picture can provide wavelength channels for near-ultraviolet, red, green, blue, near-infrared, mid-infrared, and long-infrared light. HSI consists of many smaller bands (10–20 nm). More than a hundred bands can form a hyperspectral image.



**Fig. 2** Dataset images of turmeric with tartrazine-rice flour adulterants (a) original turmeric, (b) turmeric with 5% rice flour, (c) turmeric with 10% rice flour, and (d) turmeric with 15% rice flour

## 2.3 DCNN

DCNN models, such as VGG16, DenseNet201, and MobileNet, were selected for this work due to their excellent image classification performances. These models have different architectures and layers. The following sections will discuss these DCNN models briefly.

### 2.3.1 DenseNet201

Dense convolutional networks (DenseNets) require fewer parameters than traditional CNNs since redundant maps have not been learned [28]. DenseNet has thin layers, i.e., 12 filters, adding a smaller range of new maps. Four variants are available in DenseNet, namely, DenseNet121, DenseNet169, DenseNet201, and DenseNet264. DenseNet201 is used for detection of adulterant in turmeric. Each DenseNet layer receives additional inputs from all preceding layers and transfers them to all layers. Each layer can access the original input image and gradients directly through the loss function. As a result, the calculation efficiency increased considerably, making DenseNet a better option to classify images. The basic DenseNet architecture of the network is shown in Fig. 3.

The transition layer consists of a convolution layer with a filter size of  $1 \times 1$  and an average pooling layer ( $2 \times 2$ ). The dimension of the feature map is the same in the dense block, which is helpful to combine them easily. Global average pooling was applied after the last dense block. The Softmax classifier is used to classify the images. The benefits of DensNet are strong gradient flow, computational efficiency, and more parameter-diversified features.

### 2.3.2 MobileNetV2 model

MobileNetV2 is an architecture for neural networks designed to perform on mobile systems. MobileNets are the first CV model for reliable precision in considering onboard or embedded application resource constraints. MobileNets were designed to fulfill the limitations of a few applications on the resources of compact, low power, and low latency models. They are designed for identification, detection, integration, and segmentation, similar to other large-scale models. MobileNet consists of depth-separable convolutions that enable a regular convolution into a depthwise convolution and a pointwise convolution ( $1 \times 1$ ) [14]. A single filter is used in the deep convolution layer of each input channel in the MobileNet architecture. In pointwise convolution, the outputs are integrated by a convolution size of  $1 \times 1$  into a depthwise convolution. Inputs combined with newly generated outputs in one step are coupled with regular convolution filters. The detached filtering layer and separate combinational layer

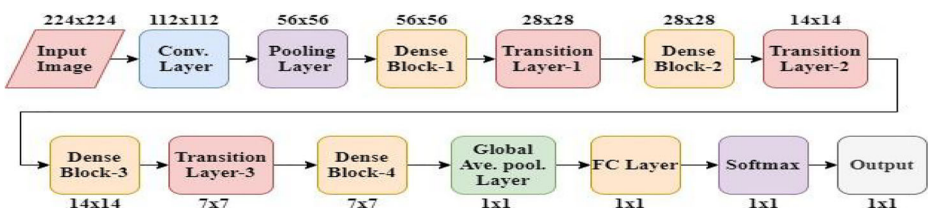


Fig. 3 Basic network architecture of the DenseNet



separate the depthwise separable convolution into two layers. Except for the first complete convolution layer, MobileNet is focused on depthwise convolutions. The MobileNet network is easy, and network topologies can be easily explored to build a successful network. Figure 4 demonstrates the basic architecture of MobileNet. Each layer is accompanied by a batch norm and a nonlinear ReLU, except for a nonlinear final FC layer.

### 2.3.3 VGG-16 model

The VGG-16 model was developed by Zisserman and SimonyanK [21], which is one of the best architecture. VGG16 architecture has  $3 \times 3$  filter with stride one in the CL and consistently employed "same" padding and Max pooling (MP) layers. CL and MP layers are structured consistently in the VGG-16 architecture. In addition, two FC layers and a softmax layer were employed for classification. Figure 5 shows the basic architecture of the VGG-16. The default input image size of the VGG-16 is  $224 \times 224$ .

### 2.4 Convolutional Neural Network (CNNs)

CNNs are one of the most important approaches for a variety of applications. The architecture of the CNN mainly consists of various neural layers: convolutional layers, batch normalization layer, pooling layers, and fully connected (FC) or flattened layers, as shown in Fig. 6.

There is a different role for each type of layer. Every layer of CNN transforms the input volume into a neuron activation output volume, which eventually leads to the complete connected layers, which results in input data mapping into a 1D vector. In CV, CNNs have been successfully applied in various applications such as facial recognition, detection of objects, robotic power visualization, and self-driving cars [66]. Network training can be split into backward and forward stages [61]. In the forward phase, the input picture classifies each layer according to its weight and bias. By using the predicted output, loss costs are calculated based on the input data. The gradients in the backward or reversed-phase are calculated for each parameter, which depends on the costs of loss measured. These parameters are updated with gradients in the next iteration. The training process stops after a good number of repetitions.

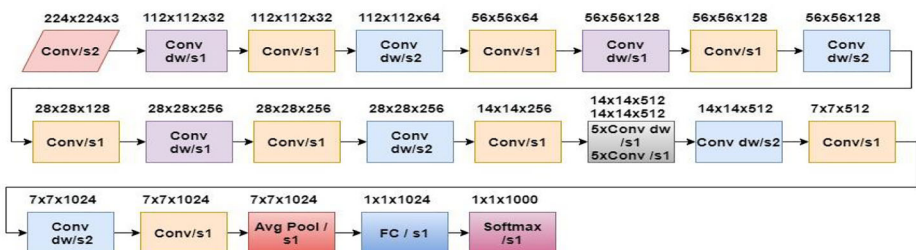


Fig. 4 Basic architecture of MobileNet



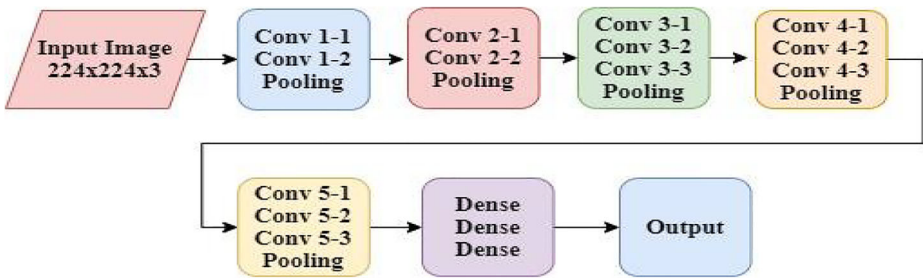


Fig. 5 Basic architecture of VGG-16

### 2.4.1 Convolutional layers(CL)

In this layer, CNN uses numerous filters to combine the entire image with intermediate function maps to generate different character maps. Due to the advantages of the operation, several projects have been proposed to replace FC layers to achieve faster learning times. The main privileges for the operation of convolution [25] are:

- Use weight-sharing mechanisms to reduce the number of parameters.
- Local connectivity makes a correlation between neighboring pixels easy.
- Ease of setting the Object location. This advantages leads users to replace FC layers to advance the learning process [51, 64].

### 2.4.2 Pooling layers

The pooling layer is similar to the convolution layer but minimizes feature map measurements and network parameters. The pooling layer is responsible for reducing the volume input for the next convolutional layer in spatial dimensions (width height). The pooling layer does not affect the volume depth. The operation performed by the pooling layer is also called subsampling or downsampling, as a simultaneous loss of information is caused by a reduction in the size. However, this loss benefits the network because the decrease in size leads to less calculative overhead for the next network layers and to an overfitting problem. The most frequently used strategies are average pooling (AP) and max pooling (MP). Detailed theoretical analysis of the

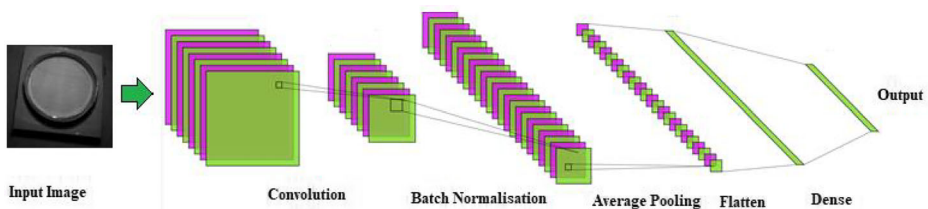


Fig. 6 Typical convolutional neural network architecture

average performance of max pooling and AP is provided [8]. AP and MP are generally used in CNN. Boureau et al. [8] have shown theoretical data on performance for average and max pooling. The pooling layer is the most extensively investigated among all three layers. These various approaches to pooling are discussed as follows:

- **Stochastic pooling** [68] - It is introduced to overcome the problem of sensitivity to overfit the training set in max pooling. The activation is assigned randomly per multinomial division in each pooling area. In this method, similar types of inputs with little variation are used to overcome the overfitting problem.
- **Spatial pyramid pooling** [24] - As a general rule, the input measurements of a picture are fixed while using CNN, and the precision of a variable picture will be compromised. The last layer of the CNN architecture could be replaced by a spatial pyramid pooling layer to remove this problem. It requires an arbitrary input and offers a flexible size, aspect ratio, and scalable solution.
- **The Def pooling** [52] - Def pooling layer is sensitive to visual pattern distortions and is used to solve computer vision deformation ambiguity.

### 2.4.3 Activation layer (ReLU)

An activation function is used to convert its inputs to outputs of a particular range. Rectified linear unit (ReLU) is one of the milestones in the DL revolution. ReLU is not only simple but also superior than earlier activation functions, such as sigmoid or tanh [47].

$$F(x)_{\text{ReLU}} = \max(0, x) \quad (1)$$

The ReLU function returns 0 if it receives a negative input and it returns x for a positive value.

### 2.4.4 Batch normalization (BN) layer

It is a way to improve the stability of the neural network by adding additional layers in a deeper, more rapid neural network. This new layer performs standardization procedures on the input of the previous layer. The normalization by batch included a transformation process, which keeps the mean output near zero and the output standard deviation near 1. Standardizing batches increases the speed of training and solves an internal problem of covariance shifts. The input is first standardized in these layers, rescaled, and then offset.

### 2.4.5 Fully connected (FC) layers

The high-level reasoning in the neural network is carried out via FC layers after several convolutions and pooling layers. Neurons in an FC layer have full connections to all activations on the previous layer. Therefore, their activation can be calculated by multiplying the matrix by offsetting the bias. Eventually, FC layers convert 2D maps to a 1D function vector. The derived vector can be fed to several classification categories [35] or considered an additional processing feature for further processing features [20]. 90% of the parameters

comprise the final layer of CNN. The feed-in network constitutes a vector with a specific length for monitoring processing. Because the majority of the parameters are in these layers, the training involves high computer stress.

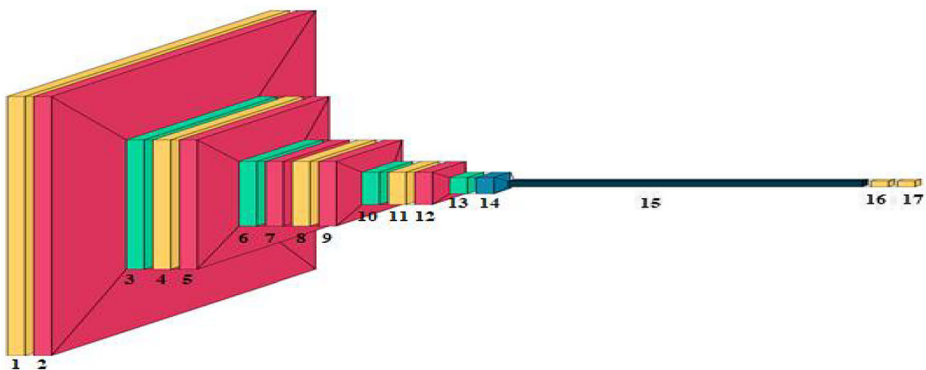
### 2.4.6 Dropout layer

The dropout layer sets input units randomly to 0, and each step during exercise frequency helps prevent overfitting. Inputs not set to 0 are scaled to  $1/(1 - \text{rate})$  to ensure that the sum remains unchanged.

## 2.5 CNN implementation

Figure 7 depicts the proposed CNN framework. The input shape of the image is  $224 \times 224 \times 3$ . The details of the CNN and FC layers are defined as follows:

- The first and second CLs begin with 8 and 16 filters, respectively, and the filter size of both layers is  $3 \times 3$ . The ReLU with padding ‘same’ is used for both layers. BN and AP are employed with a  $2 \times 2$  pool size and stride. The 20% dropout layer is applied after the AP process, and then BN is applied.
- The third CL layer has 32 filters with a  $3 \times 3$  filter size. ReLU with ‘same’ padding is employed.
- The fourth CL begin with 64 filters of  $3 \times 3$  size. ReLU with ‘same’ padding is employed. BN and AP are employed with pool sizes of  $2 \times 2$  and strides of  $2 \times 2$ . A 20% dropout is applied after the AP process, and then BN are employed.
- 64 neurons in the first FC or dense layer, followed by the ReLU function with dropout of 35%.
- Second dense layer has four neurons for four output classes.



**Fig. 7** Proposed CNN model architecture. (1- First CL, 2- BN, 3- AP, 4- Second CL, 5- BN, 6- AP, 7- BN, 8- Third CL, 9- BN, 10- AP, 11- Fourth CL, 12- BN, 13- AP, 14- Dropout layer, 15- Flatten, 16- First dense layer, 17- Second dense layer

The details of the CNN architecture with the number of layers, shape, and parameters are tabulated in Table 1.

## 2.6 Cloud computing

Cloud computing is an advancement in the area of information technology (IT) technology. Cloud computing is one of the important and dominant business models for IT service distribution. The cloud computing system is helpful to provide infrastructure to the Internet of Things (IoT), mobile computing, large data, and AI. This accelerates the dynamics of the industry, disarranges existing models, and fuels digital change [63]. Cloud computing is categorized based on its deployment and service type, as shown in Fig. 8a. There are four types of model deployment services: IaaS, PaaS, SaaS, and function as a service (FaaS). SaaS is the foremost popular cloud service, and the software resides on the provider platform. It employs a predefined protocol, which controls the services and applications [73]. PaaS is popular among the service models of cloud computing due to its capabilities in optimizing development, productivity, and business agility [69]. PaaS provides middleware resources to cloud customers [34]. It also provides development and testing environments to customers. The client creates its own application on a virtual server and manages the hosting environment for the application. The IaaS model is the basic service delivery model. It provides a broad range of on-demand virtualized IT services [9]. Here, the client rents the servers and storage of cloud service providers for building their application. FaaS is nothing but serverless computing that split cloud applications into smaller components and runs the application in accordance with the requirement. Based on the deployment, clouds are

**Table 1** Details of the implemented CNN model with the number of layers, shape, and parameters

Layer (type)	Output Shape	Param #
conv2d_32	(None, 224, 224, 16)	448
batch_normalization_45	(None, 224, 224, 16)	64
average_pooling2d_32	(None, 112, 112, 16)	0
conv2d_33	(None, 112, 112, 32)	4640
batch_normalization_46	(None, 112, 112, 32)	128
average_pooling2d_33	(None, 56, 56, 32)	0
batch_normalization_47	(None, 56, 56, 32)	128
conv2d_34	(None, 56, 56, 64)	18,496
batch_normalization_48	(None, 56, 56, 64)	256
average_pooling2d_34	(None, 28, 28, 64)	0
conv2d_35	(None, 28, 28, 128)	73,856
batch_normalization_49	(None, 28, 28, 128)	512
average_pooling2d_35	(None, 14, 14, 128)	0
dropout_24	(None, 14, 14, 128)	0
flatten_8	(None, 25,088)	0
dense_23	(None, 64)	1,605,696
dense_24	(None, 4)	260
Total params:		1,704,484

classified as private, public, and hybrid clouds. Private clouds are cloud environments devoted only to the end user and commonly run on-site or not beyond the user's firewall. These cloud services are offered to the employees of the company. The security problems are minimal in private cloud services. However, a public cloud is an external provider infrastructure that can include servers in one or more data centers. Several companies prefer public clouds in comparison to private clouds. It is less secure than a private cloud. Hybrid cloud deployments incorporate private and public clouds and include legacy on-site servers.

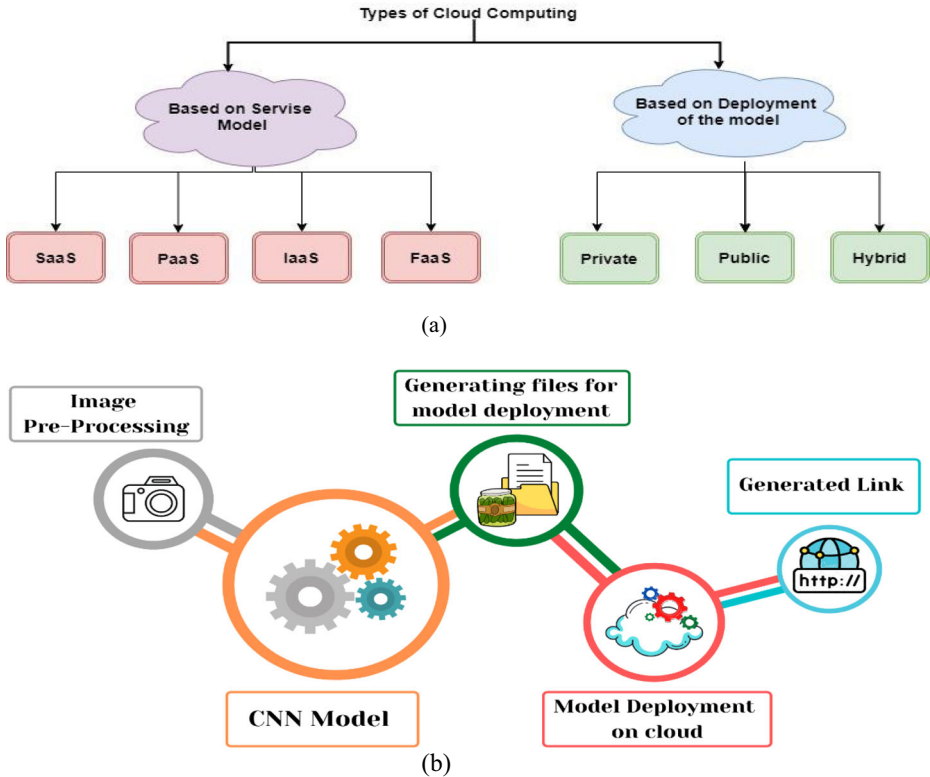
### 2.6.1 Heroku cloud

Here, the authors have used the Heroku cloud, which is an easy-to-use cloud service platform for deployment applications. Heroku has a PaaS cloud architecture and provides free service solutions for small applications. PaaS cloud architectures include operating systems, execution environments for programming languages, libraries, databases, web servers, and platform accessibility. Users of PaaS services have access to the cloud infrastructure to deploy user-created or acquired applications onto the cloud.

### 2.6.2 Cloud deployment

In the cloud deployment process, the Hypertext Markup Language (HTML) is used for the basic page setup, and Flask is a Web framework for front-end appearance. The hypertext transfer protocol (HTTP) is a protocol for sending HTML documents. It has been designed to communicate between web browsers and web servers. The internet uses it to interact and communicate with computers and servers. Flask is a Python-based web platform that provides tools and features to facilitate web application development. ML models can be deployed to clouds in various ways, such as deployment with Git, Docker-based deployments, GitHub, Heroku button, Hashicorp with Terraform, and WAR deployment. Figure 8b shows the model deployment process on cloud. The model deployment process is as follows:

- Image cleaning and resize.
- Data analysis and processing.
- Train and evaluate the model with turmeric adulterated spectral images.
- Flask module used for web server building.
- Create a template folder because Flask looks for HTML files in the template folder.
- Create files such as requirement.txt and proc files.
- Use GitHub to commit the code.
- Create a Heroku cloud account for a new project.
- Link cloud with GitHub to deploy trained CNN model. It interacts with GitHub to enable the use of GitHub code.
- Heroku can deploy the models automatically when a Heroku app configures GitHub integration. The option for manual implementation is also available.
- It can take some seconds or minutes to install all libraries in the requirement.txt file. After the installation process is completed, the HTTPS connect link will be created. The web app is ready now.
- To upload a multi spectral picture, click the above generated link. Concentrated adulteration level will appear on the mobile display immediately.



**Fig. 8** a) Types of cloud computing and b) model deployment process on cloud

### 3 Results

The dataset includes 1236 multispectral images of adulterated turmeric. The dataset was divided at a ratio of 80:20 into two parts, namely, the training (988 images) and validation (248 images) datasets. CNN and DCNN ML models have been implemented in Python. The DCNN models were pretrained on ImageNet models, which are available in Keras. The sequential method is used for creating CNN and DCNN models. ImageDataGenerator from keras.preprocessing is used to import data as well as labels. This class is helpful to rotate, rescale, flip, zoom, etc.

RMSprop and Adam optimizers are employed through the model training to attain global minima. The manual search method is used to select the values of parameters and hyperparameters. In this method, first, assign the random parameters and observe the result. Based on that result, update the parameters. This process is repeated until the parameters that will give the best accuracy. The final parameters of the Adam optimizer were  $lr=0.0003$ ,  $\beta_1=0.9$ ,  $\beta_2=0.999$ ,  $\epsilon=None$ ,  $\text{decay}=1e-8$ , and  $\text{amsgrade}=False$ . The parameters of the RMSprop optimizer were  $\text{learning\_rate} = 0.0001$ ,  $\rho = 0.99$ ,  $\epsilon = 1e-08$ ,  $\text{decay} = 0$ . The loss function categorical cross-entropy is used. The training and validation images were transferred in a batch size of 32 during the model training. The model starts the training and

shows the area under the curve (AUC), the accuracy of training, and accuracy of validation with loss. Weights are adjusted in accordance with the image validation set provided.

### 3.1 CNN accuracy and AUC

The proposed CNN model is also used to justify the various architectural choices using two optimizers, namely, Adam and RMSprop. This study is helpful for testing model performance with different parameters. The CNN with Adam and RMSprop optimizers was implemented, which achieved 100% training and 94.35% validation accuracy. The callback function was applied to terminate or stop the training process when the validation accuracy equals 94.35%. The training process with the Adam and RMSprop optimizers for the CNN model stopped at 30 and 31 epochs, respectively. Callbacks are helpful during the training process when viewing internal conditions and model statistics. The performance of the models was evaluated with data sets for training and validation.

The training accuracy, validation accuracy and AUC of the CNN models are shown in Fig. 9. Normally AUC is between 0 and 1. A model with 100% false predictions has an AUC of 0, and a model with 100% accurate predictions has an AUC of 1. In this work, the AUC is 1 for the training dataset at epochs 30 (Fig. 9e) and 31 (Fig. 9f), which means that the model predicts 100% correct results.

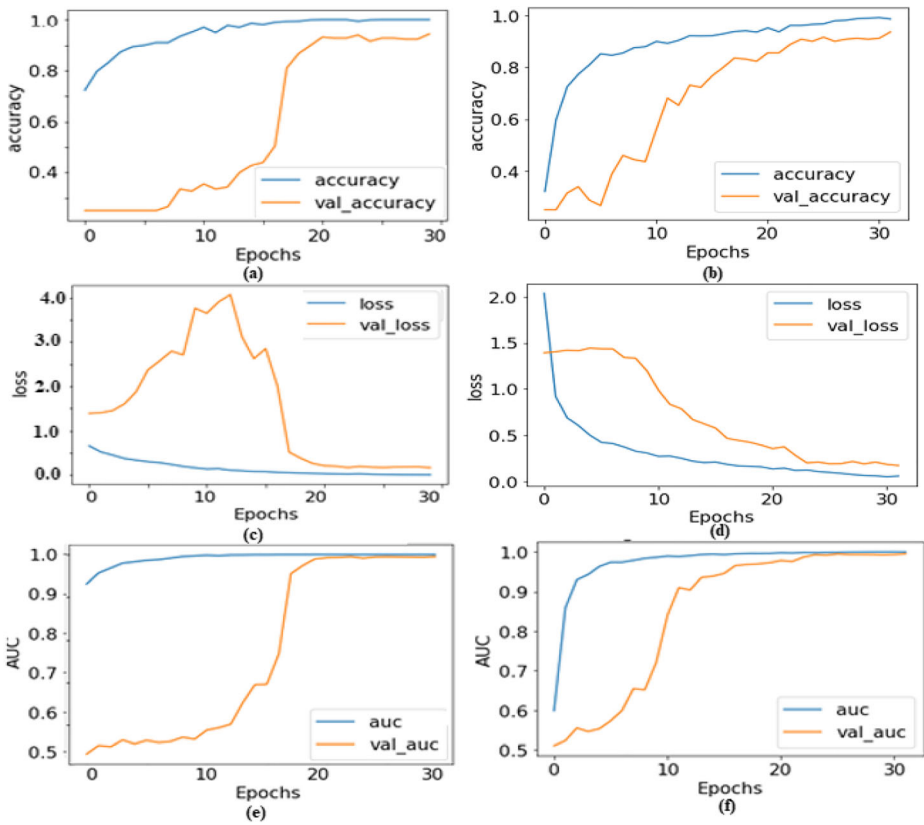
### 3.2 Output of CNN layers

Three specific notions, such as local fields of reception, links in weights, and spatial subsampling, are employed in the architecture of CNNs. Every unit in a convolutional layer receives inputs from several neighboring nodes of the last layer based upon its local reception field. The low-level characteristics were extracted as edges, curves, and lines from the first convolutional layer. The following convolutional layers have global characteristics. The CNN model uses a backpropagation procedure to understand the features. Divide the backpropagation algorithm as the forward transition, loss function, reverse transition, and weight upgrade. The CNN takes the training image through the whole network during the forward process. The weight value is initialized randomly at the start. The CNN cannot examine the low-level properties with these weights, so the CNN cannot give the right result. It returns to the network this error or loss. Figure 10 shows the output of convolutional and dense layers of CNN.

### 3.3 Comparisons of CNN with DCNN

The comparative performance analysis of the CNN and DCNN models was evaluated using a confusion matrix. The training accuracies of the proposed CNN-A, CNN-R, DenseNet201-A, DenseNet201-R, MobileNet-A, MobileNet-R, VGG16-A, and VGG16-R models are 100%, 99.09%, 99.29%, 99.70%, 99.90%, 100%, 83.20%, and 79.96%, respectively. On the other hand, the validation accuracies of the CNN-A, CNN-R, DenseNet201-A, DenseNet201-R, MobileNet-A, MobileNet-R, VGG16-A, and VGG16-R models are 94.35%, 93.55%, 89.52%, 89.92%, 91.96%, 91.53%, 79.84% and 83.87%, respectively. All the models have training accuracies above 99%, except the VGG16 model. The training accuracy of CNN-A and MobileNet-R is 100% and the highest among the models, whereas the CNN-A model achieves the highest validation accuracy of 94.35%. A comparative analysis of the CNN and DCNN



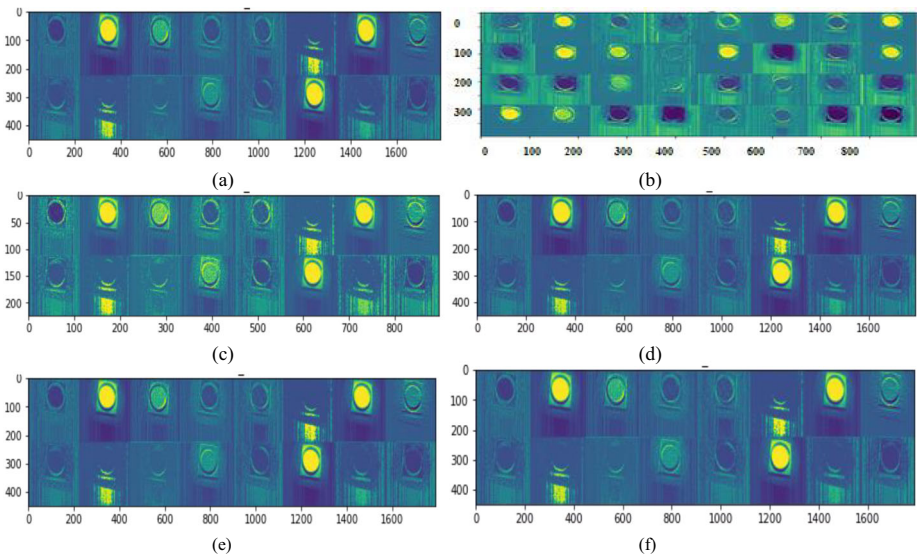


**Fig. 9** (a) Training and validation accuracy of CNN model with Adam optimizer, (b) training and validation accuracy of CNN model with RMSprop optimizer, (c) training and validation loss of CNN model with Adam optimizer, (d) training and validation loss of CNN model with RMSprop optimizer, (e) training and validation AUC of CNN model with Adam optimizer, and (f) training and validation AUC of CNN model with RMSprop optimizer

models in terms of the training and validation accuracy with loss is shown in Fig. 11. The other important parameter is the error or loss function to test any DL models. The training losses of CNN-A and CNN-R are 0.0084 and 0.0064, respectively. The validation losses of CNN-A and CNN-R are 0.1684 and 0.1179, respectively. The training loss and validation loss of the CNN-A and CNN-R models are low among the models. This performance indicated that the performance of both CNN models was excellent without overfitting.

### 3.4 ML model cross validation using confusion matrix

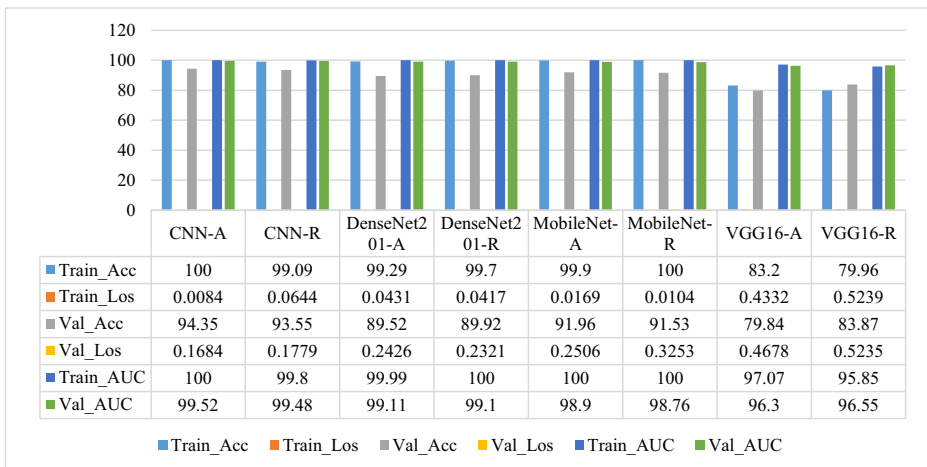
The confusion matrix was used to determine the performance of each ML model. The confusion matrix differentiates the ML model predicted values with actual values. The confusion matrix is employed to evaluate the efficacy and types of mistakes of the ML models. The confusion matrix is generated for the positive and negative classes.



**Fig. 10** (a) Output of the first CL (Conv2D-54), (b) Output of the second CL (Conv2D-55), (c) Output of the third CL (Conv2D-56), (d) Output of the fourth CL (Conv2D-57), (e) Output of the first dense layer (Dense-24), and (f) Output of the second dense layer (Dense-25)

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

where Acc-Accuracy, TP-Truly Positive, FP-False Positive, TN-Truly Negative, and FN-False Negative.



**Fig. 11** Comparative analysis of CNN and DCNN models in terms of the training and validation accuracy with loss (Train\_Acc indicates training accuracy, Val\_Acc indicates validation accuracy, Train\_Los indicates training loss, and Val\_Los indicates validation loss)

Precision (Prec): This metric shows the cases that have shown their favorable correctness.

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)$$

True positive rate (TPR) or recall (R): This metric gives the sense that the model used is correct or not correct.

$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (4)$$

F1-score (F1): F1-score offers a summary of precision and recall. It can be determined by using following Eq. 5.

$$\text{F1} = \frac{2}{\left(\frac{1}{\text{R}}\right) + \left(\frac{1}{\text{Prec}}\right)} \quad (5)$$

Specificity or True Negative Rate (TNR):

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (6)$$

### 3.5 Performance of CNN and DCNN models

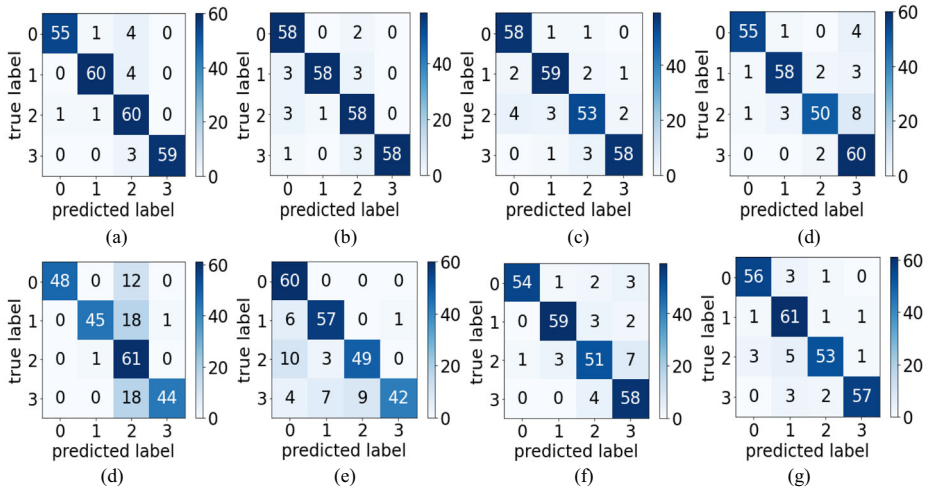
Figure 10 summarizes the performance of the proposed CNN and DCNN models of the training and validation datasets using Eqs. (2–6). Figure 12 shows the confusion matrix of correct and incorrect predictions of the CNN and DCNN models. CNN-A, CNN-R, MobileNet-A, MobileNet-R, VGG16-A, VGG16-R, DenseNet201-A, and DenseNet201-R predict 234, 232, 228, 223, 198, 208, 222, and 227 images, respectively, out of 248 validation dataset images. The performance of CNN-A outperformed that of CNN-R.

Figure 13 shows the bar charts of the performance measures, such as the accuracy, precision, F1-score, and recall of the models. The accuracy, precision, F1-score, and recall metric for the CNN-A and CNN-R models are above 0.94. The F1-score indicates the complete model performance. The highest F1-score is 0.95, achieved by the CNN-A model and outperforms all other models. Figure 14 shows the comparative analysis of CNN models with all DCNN models in terms of the true positive rate (TPR) and TNR.

From Fig. 14, it can be seen that the highest TPR of 0.94 was obtained by the CNN-A and CNN-R models. The TPR value of 0.94 indicates 94 spectral images out of 100 correctly classified by the CNN-A and CNN-R models. On the other hand, the CNN-A and CNN-R models have the highest TNR of 0.98. The TPR and TNR values should be 1 for better performance of the model. The performance of the CNN-A and CNN-R models is remarkable in terms of TPR and TNR for spectral image classification.

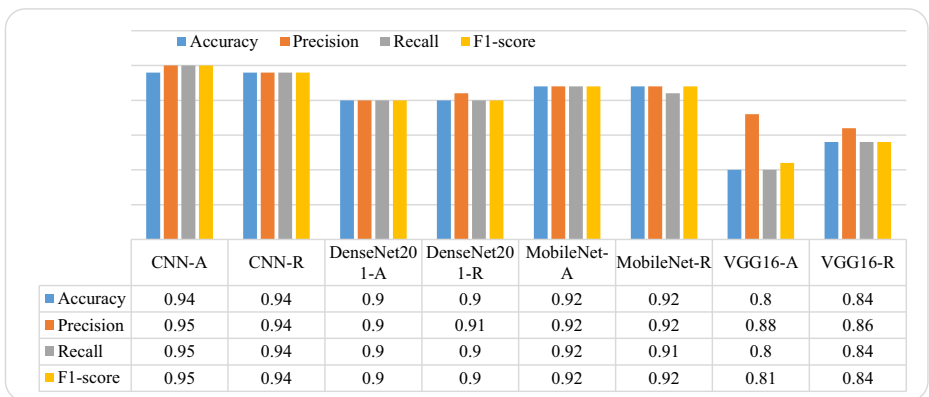
### 3.6 Cloud computing

Only a restricted platform may employ a large ML model and cannot be transferred in mobile or embedded processors. Excessive bandwidth utilization makes many consumers deeply daunting if they wish to communicate over the network. On the other hand, the large-sized



**Fig. 12** The confusion matrix depiction of classification results (a) confusion matrix of CNN-A, (b) confusion matrix of CNN-R, (c) confusion matrix of MobileNet-A, (d) confusion matrix of MobileNet-R, (e) confusion matrix of VGG16-A, (f) confusion matrix of VGG16-R, (g) confusion matrix of DenseNet201-A and (g) confusion matrix of DenseNet201-R (0-original turmeric, 1-turmeric with 10% adulterant, 2- turmeric with 15% adulterant, and 3-turmeric with 5% adulterant)

model creates enormous problems for equipment power consumption and operational speed. Due to this, practical implementation of a large model is difficult in embedded systems or mobile devices. How accurate the predictions of the excellent model are not the only factor when DL or ML is implemented on mobile devices. Many additional criteria were also considered, such as the amount of space used up by the model, how much storage it requires on an embedded device or smartphone during operation, how fast the model runs, and how fast the battery drains [26]. Researchers are frequently unconcerned about these issues. Their models may be executed on powerful desktop GPUs or computational clusters. Therefore, there is a need to develop a model with fewer parameters and a small memory size. These are important when deploying the model to the cloud because the cost is involved. The computation cost is also estimated because it helps to estimate whether computing costs are likely to



**Fig. 13** Comparative analysis of CNN and DCNN models

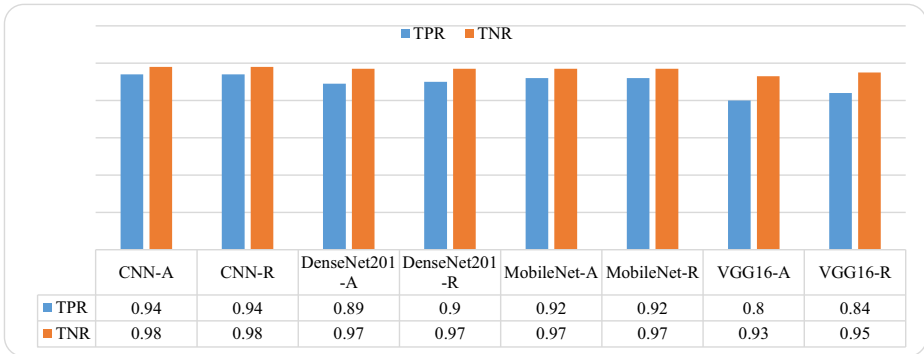


Fig. 14 TPR and TNR performance of the CNN and DCNN models

cause an override when simulating it on a real-time processor [16]. The increasing computational cost is due to increased width (numbers of filters), depth (numbers of layers), smaller strides, and their combinations [23]. A commercial search engine must respond to a request in real time, and a cloud service must handle thousands of user-submitted photographs per second. Increasing the computing capability of the hardware can help alleviate some of these issues, but it comes at a high commercial cost. Furthermore, the low computing power (CPUs or low-end GPUs) of smartphones and portable devices limits the speed of real-world recognition applications. In industrial and commercial applications, engineers and developers are frequently confronted with the demand of a tight time budget. Calculation costs are more difficult to measure because the costs vary according to the hardware, and they depend on the memory size, type of processor such as CPU, GPU, TPU, etc.

In this section, the performance of the CNN and DCNN models is assessed in terms of F1-score, layers, memory size, parameters, execution times, and prediction time. All experiments were performed on the Google Colab notebook. The comparative analysis of the CNN and DCNN models is tabulated in Table 2.

The F1-score metric which is helpful to evaluate the overall performance of the model. The F1-score of the CNN model is 95%, which is highest among the models. The CNN models have 17 layers, which are fewer layers than the DenseNet201 (201 layers), VGG16 (26 layers), and MobileNet (88 layers) models. In general, the large number of layers increases performance, but learning process is more complex. The memory size of the CNN is 19.61 MB, which is 74% smaller than DenseNet201 (74.63 MB), 65% than VGG16 (56.97 MB), and 1.3% bigger than MobileNet (14.87 MB). Moreover, the proposed CNN model has 99% fewer parameters than DenseNet201, 99% fewer parameters than VGG16, and 96% fewer

Table 2 Comparative analysis of CNN with DCNN models in terms of F1-score, layers, memory size, parameters, execution time, and prediction time

Model	F1-score	Depth	Model size	Number of parameters	Execution time in seconds (30 epochs)	Prediction time (Seconds)
DenseNet201	90%	201	74.63 MB	20,242,984	300	0.3845
VGG16	84%	26	56.97 MB	138,357,544	300	0.4919
MobileNet	92%	88	14.87 MB	4,253,864	270	0.3733
Proposed CNN	95%	17	19.61 MB	1,704,484	270	0.3945

parameters than MobileNet. The proposed CNN model is suitable for adulteration detection because of its small size, few parameters, high processing capacity, and high overall performance. The CNN model is also used for devices having small cache memory to manage high-speed applications. The execution time of the CNN is 270 seconds for 30 epochs, which is the same as the MobileNet model and lower among others. When we upload one image for classification, the proposed CNN and DCNN models take almost the same time. The CNN model predicts the image in 0.3945 seconds, whereas the MobileNet model takes a lower time of 0.3733 seconds. It is clear from the above findings that, the CNN model outperformed the other DCNN models in terms of performance.

The proposed CNN model is suitable for prediction system because of its accuracy, F1-score, small size, fewer parameters, lower execution time, and high processing capacity. CNN efficiency tests have been very effective in this work. The CNN model is suitable for deployment to the Heroku cloud. In this work, the proposed CNN model with the Adam optimizer is deployed to the cloud. The Heroku cloud link will be generated after the successful deployment of the model. Just one click on the Heroku cloud link, which is available on the mobile device, and the website opens to upload a spectral image, as shown in Fig. 15. The next step is to upload the image. The model predicts the adulteration concentration accurately after uploading the image to the cloud.

## 4 Discussions

This research examined multispectral images through CNN and DCNN models. Traditional approaches help to identify the adulteration level, but they are time-consuming. Image processing is an efficient and excellent technique for adulteration detection and classification. Several computer-based identification methods have been developed, which are discussed in the literature review. Table 3 shows a systematic comparison with other ML approaches between the current CNN model approaches. Compared to the literature mentioned in Table 3, the proposed CNN model outperformed the literature-reported models.

Bandara et al. [5] developed a multispectral imaging system that has nine spectral wavelengths ranging from 405 nm to 950 nm. A second-order polynomial was modeled and achieved  $R^2 = 0.9911$  on the training dataset and validated using an independent sample with  $R^2 = 0.9816$ . Here, the camera sensory faults and noise impact were limited by the application of image pre-processing technologies, such as dark-current removal and adaptive filtering



Fig. 15 Web page display

**Table 3** Comparison of proposed CNN model accuracy with previous studies with results

References	Model/Classifier	Accuracy
Izquierdo et al. [29]	CNN	97–100%
Bandara et al. [5]	PCA and Bhattacharyya distance analysis	$R^2 = 0.9911$
Prabhath et al. [54]	Multispectral imaging and multivariate technique	$R^2=0.9644$
Hu et al. [27]	DRIFTS combining with chemometrics	98%
Kiani et al. [32]	PCA-SVM	89–100%
Su et al. [62]	MC-PLSDA	$R^2=0.979$
Akbar et al. [1]	ANN	$R^2= 0.88$
Proposed Work	CNN	100%

approaches. Next, PCA was used as a dimension reduction step to minimize any correlated spectrum information caused by band spectral overlap. A multivariate Gaussian distribution was constructed for each class (adulteration level) and the authentic sample. Finally, the Bhattacharyya distance method was applied to classify pure and adulterated samples. In contrast, in the proposed work, image pre-processing work is automatically performed in the CNN model itself. This means that there is no need to implement these algorithms. Principal components in PCA attempt to cover the most variance among the features in a dataset; however, if the number of principal components is not carefully chosen, it may lose some information when compared to the original list of features. The Bhattacharyya distance method can be applied only to two-class problems. Due to this limitation, Bhandara et al. classified only two classes: pure and adulterated samples. However, in the proposed work, we classified not just pure and adulterated samples but improved the classification criteria by varying the percentage (%) of adulterant (0, 5, 10, and 15%) with an adulteration detection accuracy of 94.35% during validation. In addition, successful deployment of the developed model to clouds has also been achieved.

The four classes of rice flour-adulterated turmeric multispectral images were classified using the CNN and DCNN models. The training accuracy of the CNN is 100%, which is remarkable. The CNN models are small in size and deployed to the PaaS cloud. The proposed CNN model can differentiate adulterated images to correctly identify adulteration levels in turmeric. This CNN-based system outperformed all other work, which are available in the literature. The empirical assessment shows that CNN-A models have better overall performance.

## 5 Conclusions

In this paper, CNN and DCNN models were successfully implemented using the Python programming language to classify multispectral images of toxic tartrazine-colored rice flour-adulterated turmeric. The experimental results shows promising classification accuracy with the CNN and DCNN models, but the CNN with the Adam optimizer (CNN-A) model outperformed the other models. A different set of parameters was employed in the evaluation of the CNN and DCNN models. Two optimizers, namely, Adam and RMSprop, were applied to compare the robustness of the model. The performance of the CNN model is remarkable in terms of accuracy, precision, recall, and F1-score. Furthermore, the size of the models is also less compared to the literature-reported CNN and DCNN models. The size of the model plays



an important role in cloud deployment. An accurate CNN-A model with a small size among the models is successfully deployed to the cloud. The generated deployed model link was tested on a smartphone, and it performed well. The performance analysis shows that the proposed CNN model has fewer parameters, a small memory size, 100% training accuracy, 94.35% validation accuracy, and instant results. In future work, it will be planned to use the CNN model to study other adulterants with a cost-effective approach without using complicated systems and costly equipment. This research work is further expanding to identify more adulterants at various concentration levels in the future. Despite CNN's important performance in adulteration detection, new algorithms are still enormously likely to be utilized to improve CNN's calculation speed, and further research is needed in this area. The CNN can be used to detect other adulterants, such as lead chromate and Metanil yellow. The optimized models can be deployed on the cloud, and the model link will be accessible on a smartphone.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s11042-022-12392-3>.

**Author contribution** Mr. Madhusuan Lanjewar<sup>1</sup>: Methodology, Software, Validation, Investigation, Writing - Original Draft.

Dr. Pranay P. Morajkar<sup>2</sup>: Conceptualization, visualization, corrections, modifications and rewriting of the original manuscript.

Dr. Jivan Parab<sup>\*3</sup>: Methodology, Investigation, Performance analysis, and modifications of the original manuscript.

## References

1. Akbar A, Kuanar A, Patnaik J et al (2018) Application of Artificial Neural Network modeling for optimization and prediction of essential oil yield in turmeric (*Curcuma longa* L.). *Comput Electron Agric* 148:160–178. <https://doi.org/10.1016/j.compag.2018.03.002>
2. Amani M, Kakooei M, Moghimi A, Ghorbanian A, Ranjgar B, Mahdavi S, Davidson A, Fiset T, Rollin P, Brisco B, Mohammadzadeh A (2020) Application of Google earth engine cloud computing platform, sentinel imagery, and neural networks for crop mapping in Canada. *Remote Sens* 12:3561. <https://doi.org/10.3390/rs12213561>
3. Ashok V, Agrawal N, Durgbanshi A, Esteve-Romero J, Bose D (2015) A novel micellar chromatographic procedure for the determination of metanil yellow in foodstuffs. *Anal Methods* 7:9324–9330. <https://doi.org/10.1039/C5AY02377G>
4. Bandara C (2019) Multispectral images of adulterated turmeric powder [Calibration Data]. <https://data.mendeley.com/datasets/b7cwddkcmj/3>; <https://doi.org/10.17632/b7cwddkcmj.3>
5. Bandara WGC, Prabhath GWK, Dissanayake DWSCB, Herath VR, Godaliyadda GMRI, Bandara Ekanayake MP, Demini D, Madhujith T (2020) Validation of multispectral imaging for the detection of selected adulterants in turmeric samples. *J Food Eng* 266:109700. <https://doi.org/10.1016/j.jfoodeng.2019.109700>
6. Bertelli D, Plessi M, Sabatini A, Lolli M, Grillenzoni F (2007) Classification of Italian honeys by mid-infrared diffuse reflectance spectroscopy (DRIFTS). *Food Chem* 101:1565–1570. <https://doi.org/10.1016/j.foodchem.2006.04.010>
7. Bhowmik D, Chiranjib KKPS, Chandira M, Jayakar B. Direct CAB. <https://www.cabdirect.org/?target=%2fcabdirect%2fabstract%2f20103252001>. Accessed 16 Aug 2021
8. Boureau Y-L, Ponce J, LeCun Y (2010) A theoretical analysis of feature pooling in visual recognition. In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, Haifa, pp 111–118
9. Chawki EB, Ahmed A, Zakariae T (2018) IaaS cloud model security issues on behalf cloud provider and user security behaviors. *Procedia Comput Sci* 134:328–333. <https://doi.org/10.1016/j.procs.2018.07.180>
10. Chen L, Hu J, Zhang W, Zhang J, Guo P, Sun C (2015) Simultaneous determination of nine banned azo dyes in foodstuffs and beverages by high-performance capillary electrophoresis. *Food Anal Methods* 8:1903–1910. <https://doi.org/10.1007/s12161-014-0074-6>

11. Dhakal S, Chao K, Schmidt W, Qin J, Kim M, Chan D (2016) Evaluation of turmeric powder adulterated with metanil yellow using FT-Raman and FT-IR Spectroscopy. *Foods* 5:36. <https://doi.org/10.3390/foods5020036>
12. Di Anibal CV, Odena M, Ruisánchez I, Callao MP (2009) Determining the adulteration of spices with Sudan I-II-III-IV dyes by UV–visible spectroscopy and multivariate classification techniques. *Talanta* 79: 887–892. <https://doi.org/10.1016/j.talanta.2009.05.023>
13. Di Anibal CV, Ruisánchez I, Callao MP (2011) High-resolution <sup>1</sup>H Nuclear Magnetic Resonance spectrometry combined with chemometric treatment to identify adulteration of culinary spices with Sudan dyes. *Food Chem* 124:1139–1145. <https://doi.org/10.1016/j.foodchem.2010.07.025>
14. Dong T, Liu J, Shang J, Qian B, Huffman T, Zhang Y, Champagne C, Daneshfar B (2016) Assessing the impact of climate variability on cropland productivity in the canadian prairies using time series MODIS FAPAR. *Remote Sens* 8:281. <https://doi.org/10.3390/rs8040281>
15. Ennis R, Schiller F, Toscani M, Gegenfurtner KR (2018) Hyperspectral database of fruits and vegetables. *J Opt Soc Am A* 35:B256. <https://doi.org/10.1364/JOSAA.35.00B256>
16. Estimate Computation Costs - MATLAB &, Simulink. <https://www.mathworks.com/help/physmod/simscape/ug/estimate-computation-costs.html>. Accessed 17 Aug 2021
17. Fadda E, Manerba D, Cabodi G et al (2021) Comparative analysis of models and performance indicators for optimal service facility location. *Transp Res E* 145:102174. <https://doi.org/10.1016/j.tre.2020.102174>
18. Fadda E, Manerba D, Cabodi G et al (2021) Evaluation of Optimal Charging Station Location for Electric Vehicles: An Italian Case-Study. In: Fidanova S et al (eds) *Recent Advances in Computational Optimization*. Springer International Publishing, Cham, pp 71–87
19. Fuh M (2002) Determination of sulphonated azo dyes in food by ion-pair liquid chromatography with photodiode array and electrospray mass spectrometry detection. *Talanta* 56:663–671. [https://doi.org/10.1016/S0039-9140\(01\)00625-7](https://doi.org/10.1016/S0039-9140(01)00625-7)
20. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs]
21. Guan Q, Wang Y, Ping B, Li D, Du J, Qin Y, Lu H, Wan X, Xiang J (2019) Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study. *J Cancer* 10:4876–4882. <https://doi.org/10.7150/jca.28769>
22. Hatcher H, Planalp R, Cho J, Torti FM, Torti SV (2008) Curcumin: From ancient medicine to current clinical trials. *Cell Mol Life Sci* 65:1631–1652. <https://doi.org/10.1007/s00018-008-7452-4>
23. He K, Sun J (2014) Convolutional neural networks at constrained time cost. arXiv:14121710 [cs]
24. He K, Zhang X, Ren S, Sun J (2014) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) *Computer Vision – ECCV 2014*. Springer International Publishing, Cham, pp 346–361
25. Hierarchical Convolutional Deep Learning in Computer Vision - ProQuest. <https://www.proquest.com/openview/62c046242f67ce115a76b9224e66a69c/1?cbi=18750&diss=y&pq-origsite=gscholar>. Accessed 17 Aug 2021
26. How fast is my model? <https://machinethink.net/blog/how-fast-is-my-model/>. Accessed 17 Aug 2021
27. Hu L, Yin C, Ma S, Liu Z (2018) Assessing the authenticity of black pepper using diffuse reflectance mid-infrared Fourier transform spectroscopy coupled with chemometrics. *Comput Electron Agric* 154:491–500. <https://doi.org/10.1016/j.compag.2018.09.029>
28. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2018) Densely connected convolutional networks. arXiv:1608.06993 [cs]
29. Izquierdo M, Lastra-Mejías M, González-Flores E, Cancilla JC, Aroca-Santos R, Torrecilla JS (2020) Deep thermal imaging to compute the adulteration state of extra virgin olive oil. *Comput Electron Agric* 171: 105290. <https://doi.org/10.1016/j.compag.2020.105290>
30. Jayaprakasha GK, Jagan Mohan Rao L, Sakariah KK (2002) Improved HPLC method for the determination of curcumin, demethoxycurcumin, and bisdemethoxycurcumin. *J Agric Food Chem* 50:3668–3672. <https://doi.org/10.1021/jf025506a>
31. Khodabakhshian R, Emadi B, Khojastehpour M, Golzarian MR (2017) Determining quality and maturity of pomegranates using multispectral imaging. *J Saudi Soc Agric Sci* 16:322–331. <https://doi.org/10.1016/j.jssas.2015.10.004>
32. Kiani S, Minaei S, Ghasemi-Varnamkhasti M (2017) Integration of computer vision and electronic nose as non-destructive systems for saffron adulteration detection. *Comput Electron Agric* 141:46–53. <https://doi.org/10.1016/j.compag.2017.06.018>
33. Kiani S, van Ruth SM, Minaei S, Ghasemi-Varnamkhasti M (2018) Hyperspectral imaging, a non-destructive technique in medicinal and aromatic plant products industry: Current status and potential future applications. *Comput Electron Agric* 152:9–18. <https://doi.org/10.1016/j.compag.2018.06.025>

34. Kim D, Schaffer HE, Vouk MA (2017) About PaaS security. *IJCC* 6:325. <https://doi.org/10.1504/IJCC.2017.090200>
35. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet classification with deep convolutional neural networks. *Commun ACM* 60:84–90. <https://doi.org/10.1145/3065386>
36. Kumar N, Dahiya AK, Kumar K (2020) Image Restoration Using a Fuzzy-Based Median Filter and Modified Firefly Optimization Algorithm. *Int J Adv Sci Technol* 29:1471–1477
37. Kumar N, Dahiya AK, Kumar K (2020) Modified Median Filter for Image Denoising. *Int J Adv Sci Technol* 29:1495–1502
38. Kwan C (2019) Methods and challenges using multispectral and hyperspectral images for practical change detection applications. *Information* 10:353. <https://doi.org/10.3390/info10110353>
39. Lan H, Zheng X, Torrens PM (2018) Spark sensing: a cloud computing framework to unfold processing efficiencies for large and multiscale remotely sensed data, with examples on Landsat 8 and MODIS Data. *J Sens* 1–12. <https://doi.org/10.1155/2018/2075057>
40. Lee B-H, Dewi EK, Wajdi MF (2018) Data security in cloud computing using AES under HEROKU cloud. In: 2018 27th Wireless and Optical Communication Conference (WOCC). IEEE, Hualien, pp 1–5
41. Lee K, Silva BN, Han K (2021) Algorithmic implementation of deep learning layer assignment in edge computing based smart city environment. *Comput Electr Eng* 89:106909. <https://doi.org/10.1016/j.compeleceng.2020.106909>
42. Liu C, Hao G, Su M, Chen Y, Zheng L (2017) Potential of multispectral imaging combined with chemometric methods for rapid detection of sucrose adulteration in tomato paste. *J Food Eng* 215:78–83. <https://doi.org/10.1016/j.jfoodeng.2017.07.026>
43. Liu J, Zhou X, Huang J, Liu S, Li H, Wen S, Liu J (2017) Semantic classification for hyperspectral image by integrating distance measurement and relevance vector machine. *Multimed Syst* 23:95–104. <https://doi.org/10.1007/s00530-015-0455-8>
44. Malapela T. Is there a potential in adopting Artificial Intelligence in food and agriculture sector, and can it transform food systems and with what impact? | E-Agriculture. <http://www.fao.org/e-agriculture/news/there-potential-adopting-artificial-intelligence-food-and-agriculture-sector-and-can-it>. Accessed 17 Aug 2021
45. McNairn H, Brisco B (2004) The application of C-band polarimetric SAR for agriculture: a review. *Can J Remote Sens* 30:525–542. <https://doi.org/10.5589/m03-069>
46. Morajkar PP, Naik AP, Bugde ST, Naik BR (2019) Photocatalytic and microbial degradation of Amaranth dye. *Advances in Biological Science Research*. Elsevier, Amsterdam, pp 327–345
47. Mujtaba H. An introduction to Rectified Linear Unit (ReLU) | What is ReLU? <https://www.mygreatlearning.com/blog/relu-activation-function/>. Accessed 16 Aug 2021
48. Naik AP, Salkar AV, Majik MS, Morajkar PP (2017) Enhanced photocatalytic degradation of Amaranth dye on mesoporous anatase TiO<sub>2</sub>: evidence of C–N, NN bond cleavage and identification of new intermediates. *Photochem Photobiol Sci* 16:1126–1138. <https://doi.org/10.1039/C7PP00090A>
49. Naik AP, Sawant JV, Mittal H, Al Alili A, Morajkar PP (2021) Facile synthesis of 2D nanoflakes and 3D nanosponge-like Ni<sub>1–x</sub>O via direct calcination of Ni (II) coordination compounds of imidazole and 4-nitrobenzoate: Adsorptive separation kinetics and photocatalytic removal of Amaranth dye contaminated wastewater. *J Mol Liquids* 325:115235. <https://doi.org/10.1016/j.molliq.2020.115235>
50. Naz S, Ashraf A, Zaib A (2021) Transfer learning using freeze features for Alzheimer neurological disorder detection using ADNI dataset. *Multimed Syst*. <https://doi.org/10.1007/s00530-021-00797-3>
51. Oquab M, Bottou L, Laptev I, Sivic J (2015) Is object localization for free? - Weakly-supervised learning with convolutional neural networks. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Boston, pp 685–694
52. Ouyang W, Wang X, Zeng X, Qiu S, Luo P, Tian Y, Li H, Yang S, Wang Z, Loy C-C, Tang X (2015) DeepID-Net: Deformable deep convolutional neural networks for object detection. arXiv:1412.5661 [cs]
53. Parvathy VA, Swetha VP, Sheeja TE, Sasikumar B (2015) Detection of plant-based adulterants in turmeric powder using DNA barcoding. *Pharm Biol* 53:1774–1779. <https://doi.org/10.3109/13880209.2015.1005756>
54. Prabhath GWK, Bandara WGC, Dissanayake DWSCB, Hearath HMVR, Godaliyadda GMRI, Ekanayake MPB, Demini SMD, Madhujith T (2019) Multispectral imaging for detection of adulterants in turmeric powder. *Optical Sensors and Sensing Congress (ES, FTS, HISE, Sensors)*. p. HTu3B.3. OSA, San Jose
55. Product Profiles of TURMERIC. <http://apeda.in/agriexchange/Market%20Profile/one/TURMERIC.aspx>. Accessed 14 Aug 2021
56. Ropodi AI, Panagou EZ, Nychas G-JE (2017) Multispectral imaging (MSI): A promising method for the detection of minced beef adulteration with horsemeat. *Food Control* 73:57–63. <https://doi.org/10.1016/j.foodcont.2016.05.048>

57. Salmerón-García JJ, van den Dries S, Díaz-del-Río F, Morgado-Estevez A, Sevillano-Ramos JL, van de Molengraaf MJG (2019) Towards a cloud-based automated surveillance system using wireless technologies. *Multimedia Syst* 25:535–549. <https://doi.org/10.1007/s00530-017-0558-5>
58. Sha O, Zhu X, Feng Y, Ma W (2014) Determination of sunset yellow and tartrazine in food samples by combining ionic liquid-based aqueous two-phase system with high performance liquid chromatography. *J Anal Methods Chem* 1–8. <https://doi.org/10.1155/2014/964273>
59. Shafiee S, Polder G, Minaei S, Moghadam-Charkari N, van Ruth S, Kuś PM (2016) Detection of honey adulteration using hyperspectral imaging. *IFAC-PapersOnLine* 49:311–314. <https://doi.org/10.1016/j.ifacol.2016.10.057>
60. Shah R (2017) Identification and estimation of non-permitted food colours (metanil yellow and aniline dyes) in turmeric powder by rapid color test and thin layer chromatography. *WJPPS*, 2034–2045. <https://doi.org/10.20959/wjpps20178-9867>
61. Sinha RK, Pandey R, Pattnaik R (2018) Deep learning for computer vision tasks: A review. arXiv: 1804.03928 [cs]
62. Su W-H, Sun D-W (2016) Potential of hyperspectral imaging for visual authentication of sliced organic potatoes from potato and sweet potato tubers and rapid grading of the tubers according to moisture proportion. *Comput Electron Agric* 125:113–124. <https://doi.org/10.1016/j.compag.2016.04.034>
63. Sunyaev A (2020) Cloud computing. Internet computing. Springer International Publishing, Cham, pp 195–236
64. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Boston, pp 1–9
65. Tateo F, Bononi M (2004) Fast determination of Sudan I by HPLC/APCI-MS in hot chilli, spices, and oven-baked foods. *J Agric Food Chem* 52:655–658. <https://doi.org/10.1021/jf030721s>
66. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E (2018) Deep learning for computer vision: a brief review. *Comput Intell Neurosci*: 1–13. <https://doi.org/10.1155/2018/7068349>
67. Wu L (2021) Analysis of food Additives. In: *Innovative Food Analysis*. Elsevier, Amsterdam, pp 157–180
68. Wu H, Gu X (2015) Max-pooling dropout for regularization of convolutional neural networks. arXiv: 1512.01400 [cs]
69. Xu P, Hu R, Su S (2013) Research on resource management in PaaS based on IaaS environment. In: Su J, Zhao B, Sun Z, Wang X, Wang F, Xu K (eds) *Frontiers in Internet Technologies*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 145–157
70. Yang X, Luo N, Tan Z, Jia Z, Liao X (2017) A fluorescence probe for tartrazine detection in foodstuff samples based on fluorescence resonance energy transfer. *Food Anal Methods* 10:1308–1316. <https://doi.org/10.1007/s12161-016-0691-3>
71. Zhang L, Yong W, Liu J, Wang S, Chen Q, Guo T, Zhang J, Tan T, Su H, Dong Y (2015) Determination of dicyandiamide in powdered milk using direct analysis in real time quadrupole time-of-flight tandem mass spectrometry. *J Am Soc Mass Spectrom* 26:1414–1422. <https://doi.org/10.1007/s13361-015-1142-x>
72. Zhao S, Yin J, Zhang J, Ding X, Wu Y, Shao B (2012) Determination of 23 dyes in chili powder and paste by high-performance liquid chromatography–electrospray ionization tandem mass spectrometry. *Food Anal Methods* 5:1018–1026. <https://doi.org/10.1007/s12161-011-9337-7>
73. Zheng X, Fu M, Chugh M (2017) Big data storage and management in SaaS applications. *J Commun Inf Netw* 2:18–29. <https://doi.org/10.1007/s41650-017-0031-9>
74. Zoughi S, Faridbod F, Amiri A, Ganjali MR (2021) Detection of tartrazine in fake saffron containing products by a sensitive optical nanosensor. *Food Chem* 350:129197. <https://doi.org/10.1016/j.foodchem.2021.129197>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.