IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2022, accepted 18 December 2022, date of publication 27 December 2022, date of current version 5 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3232564

RESEARCH ARTICLE

Protein Subcellular Localization Prediction by Concatenation of Convolutional Blocks for Deep Features Extraction From Microscopic Images

SONAM AGGARWAL[1], SAPNA JUNEJA[2], JUNAID RASHID[3], DEEPALI GUPTA[1], SHEIFALI GUPTA[1], AND JUNGEUN KIM[3,4]

[1]Chitkara University Institute of Engineering and Technology, Chitkara University, Chandigarh, Punjab 140401, India
[2]Department of Computer Science, KIET Group of Institutions, Ghaziabad 201206, India
[3]Department of Computer Science and Engineering, Kongju National University, Cheonan 31080, South Korea
[4]Department of Software, Kongju National University, Cheonan 31080, South Korea

Corresponding authors: Junaid Rashid (junaid.rashid@kongju.ac.kr) and Jungeun Kim (jekim@kongju.ac.kr)

This work was supported in part by the Technology Development Program of MSS under Grant S3033853; and in part by the National University Development Project by the Ministry of Education, in 2022.

ABSTRACT Understanding where proteins are located within the cells is essential for proteomics research. Knowledge of protein subcellular location aids in early disease detection and drug targeting treatments. Incorrect localization of proteins can interfere with the functioning of cells and leads to illnesses like cancer. Technological advances have enabled computational methods to detect protein's subcellular location in living organisms. The advent of high-quality microscopy has led to the development of image-based prediction algorithms for protein subcellular localization. Confocal microscopy, which is used by the Human Protein Atlas (HPA), is a great tool for locating proteins. HPA database comprises millions of images which have been procured using confocal microscopy and are annotated with single as well as multi-labels. However, the multi-instance nature of the classification task and the low quality of the images make image-based prediction an extremely difficult problem. There are probably just a few algorithms for automatically predicting protein localization, and most of them are limited to single-label classification. Therefore, it is important to develop a satisfactory automatic multi-label HPA recognition system. The aim of this research is to design a model based on deep learning for automatic recognition system for classifying multi-label HPA. Specifically, a novel Convolutional Neural Network design for classifying protein distribution across 28 subcellular compartments has been presented in this paper. Extensive experiments have been done on the proposed model to achieve the best results for multilabel classification. With the proposed CNN framework as F1-score of 0.77 was achieved which outperformed the latest approaches.

INDEX TERMS Deep learning, convolutional neural network, biomedical image analysis, protein subcellular localization prediction, proteomics.

I. INTRODUCTION

A cell's internal structure is complex but highly organized and can be separated into many subcellular sections or organelles. Different regions or organelles are dispersed throughout the cellular area and perform distinct activities. These organelles are also known as subcellular compartments. They are the essential components of the cell's complicated structure and serve as the chief executive officers of cellular processes.

A vital part of the cells within living organisms is protein. Understanding all of the functions of proteins is currently a major objective in the realm of biological sciences. This gives important information regarding the function of proteins both inside and outside the cell [1]. It is generally agreed that subcellular localization is the most important feature of proteins. The term ''protein subcellular localization'' describes

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

the process by which a gene product or protein is delivered to a certain compartment of the cell. It provides significant insight into the structural, dynamic, and functional properties of a variety of proteins [2]. Knowing where in the cell a certain protein is located can shed light on its many roles [3]. For instance, knowing where proteins are located within the cell might shed light on how cells respond in certain conditions. Furthermore, understanding the precise localization of proteins could aid in testing the efficacy of drugs [4], [5], [6]. And also, by accurately finding where proteins are in cells, it might be possible to diagnose diseases early and treat them successfully [7].

Due to technological advancements, researchers now have access to massive medical image databases for study [1]. Fluorescence microscopy is a popular tool in the biological sciences for visualizing protein localization within cells. Typically, conventional methods of analysis are used to classify these images. Traditional methods, however, are laborious, pricey, and prone to mistakes [1], [8]. Conclusions regarding the subcellular localization of proteins could be made that were inaccurate and ambiguous. On the other hand, automated classification algorithms can often outperform human inspections [9]. Therefore, automated classification techniques are required to predict protein localization [7] accurately.

Through antibody labelling and microscopy, the Human Protein Atlas (HPA) has created an image-based atlas that specifies the localization of proteins in human tissues and cells. [10]. Eukaryotic cells rely on subcellular compartmentalization as a fundamental organizing principle since it allows for the simultaneous execution of many cellular processes. Utilizing tens of thousands of high-resolution confocal immunofluorescent images, HPA is creating a proteome-scale map of subcellular location of protein [11]. Researchers can use this map to learn more about protein roles, networks, cellular biology, and even how disease originates. Since the HPA Cell Atlas is constantly collecting images, a thorough study of the data will necessitate a huge number of precise image classifications.

A high throughput microscope screen requires automated processing of the data collected. Common steps include adjusting the brightness and contrast of images, detecting and segmenting cells, obtaining features, and analyzing the data. However, while segmentation and normalization can be conducted in a somewhat consistent fashion to yield protein abundances, but feature extraction for a particular problem and statistical analysis are required for mapping subcellular localizations. Calculating more abstract features from raw pixels and selecting the most useful ones is a crucial step in any image analysis pipeline if the results are to have any practical significance [3], [12]. Defining the right features can be tedious and prone to mistakes, and the default numbers generated by existing tools may be irrelevant outside the domain for which they were designed [13], [14].

Due to its ability to solve the issue of feature selection, deep neural networks have recently gained traction in image analysis. Models trained with deep learning techniques already outperform humans at many tasks, including object detection [15], semantic segmentation [16], and image captioning [17]. These techniques have also been successfully applied to biological domains [18], [19], [20], including regulatory genomics [21], [22], [23] and electron microscopy [3], [24]. Briefly, deep networks are trained to predict observed labels by running images through multiple layers of processing units that quantify progressively more complex patterns in the data. One of their primary selling points is that, with a sufficiently big training set, they may learn, on their own, which features are most effective in resolving a given classification problem.

Therefore, in this research, a neural network model, specifically a CNN trained from scratch, has been proposed to learn the patterns of proteins in the cell and accurately predict protein location in subcellular compartments. Following are the main contributions to this paper.

1. A convolutional neural network trained from scratch has been presented to predict the protein localization in 28 subcellular compartments using confocal microscopy images procured from Human Protein Atlas database.
2. Rigorous testing is done on the proposed model with different filter sizes and different input image sizes to achieve the best results.
3. The model is evaluated based on recall, precision and F1-score and also comparison is made with the state-of-art.

The rest of the paper is presented as: In section 2, related work has been given in the paper related to machine learning and deep learning techniques. Section 3 starts with the description of the dataset used in this research, followed by the proposed CNN model. Experimental setup used for this study has also been explained in this section. Section 4 covers the results obtained from the model on different filter sizes and different input image sizes. This is followed by the visualization of results in the form of incorrect and correct predictions. Finally, the paper concludes with a conclusion and the future scope of the research.

II. RELATED WORK

Researchers have used various algorithms based on deep learning and machine learning to determine the subcellular localization and distribution of proteins in the human cells. Over the past decade, numerous machine learning-based approaches have been developed [25], [26] to assess the localization of proteins in cultured cells by combining imaging and pattern recognition. These techniques use microscopic pictures to extract subcellular location features, which are subsequently examined for patterns [6]. Not only do SLFs account for morphological details, but they also provide a quantitative description of proteins' subcellular localization. Therefore, once the protein patterns have been extracted from the images, SLFs can be used to teach classifiers to differentiate between them.

TABLE 1. Summary of literature review.

Reference / Year	Technique Used	Dataset Used	Number of Images	Results
Su et al. [27]	Feature extraction using VGGNet, ResNet, AlexNet, DenseNet and Xception followed by SVM classifier	HPA	3129 IHC images	F1-score VGGNet: 0.82 DenseNet: 0.69 ResNet: 0.73 AlexNet: 0.83 Xception: 0.40
Liu et al. [28]	Attention mechanism	HPA	3129 IHC images	F1-score: 0.524
Kraus et al. [29]	CNN	YeastFGP	21,882 IF images	Accuracy: 72.3%
Pärnamaa et al. [30]	CNN	YeastFGP	20,000 IF images	Accuracy: 87%
Xiang et al. [31]	CNN	HPA	50,864 IF images	F1-score: 0.823
Sulvian et al. [32]	CNN	HPA	5,416 IF images	F1-score: 0.72
Liimatainen et al [33]	CNN	HPA	5,416 IF images	F1-score 0.51
Liimatainen et al. [34]	CNN and FCN	HPA	20,000 IF images	F1-score CNN: 0.676 FCN: 0.696
Li et al. [35]	InceptionV3	HPA	31,072 IF images	F1-score: 0.706
Shwetha et al. [36]	Hybrid Xception	HPA	14,094 IF images	Recall: 0.69 Precision: 0.69 F1-score: 0.69

The problem of PSL prediction has been efficiently tackled by machine learning methods, although obtaining features from images is laborious. Because deep learning enables the system to learn visual information on its own, it is no longer necessary to extract these features beforehand. Recently, CNN-based techniques have been effectively employed to classify protein subcellular localization.

Recent publications have used convolutional neural networks (CNNs) to analyse data from high-throughput microscopy [27], [28], [29], [30]; however, the unique nature of HTI data makes it difficult to use generic models, due to its very high resolution. Consequently, most methods rely on data pre-processing procedures and are limited to analyzing single-cell crops, or segmented images of a single cell. Xiang et al. [31] proposed a convolutional neural network for multilabel classification of protein subcellular localization in 28 compartments and achieved an F1-score of 0.823. Immunofluorescence images from the HPA dataset were classified using a two-pronged technique by Sullivan et al. [32]. They began by holding a competition for image categorization through online video game, which yielded millions of annotations of protein locations. The data from the competition was then used to train an automated software called Loc-CAT, which then classified the protein sites into 29 sub-compartments. To categorise protein patterns, the two methods were combined and 0.72 F1-score of was obtained with combined model. To assess protein localization in yeast, another CNN model with 11 yaers was built by Kraus et al. [29] achieving an accuracy of 72.3% for proteins localised to 10 subcellular locations. Liimatainen et al. [33] have employed semantic segmentation networks for localising proteins, and they have established an F1-score of 0.51 as the state-of-the-art for this endeavour. Overall, cell segmentation techniques and data pre-processing still play significant roles in HTI analysis. Liimatainen et al. [34] presented two novel CNN-based algorithms to predict protein localization in 13 compartments achieving 0.696 and 0.676 F1-score. Li et al. [35] developed a transfer learning method by using a fine-tuned InceptionV3 architecture which gave 0.706 F1-score. Microscopic images from the HPA database were classified using two different methods by T.R. Shwetha et al. [36]. The first approach involved extracting features and classifying them with a Random Forests classifier. Another method used two different transfer learning models namely Xception and ResNet 50 for feature extraction and made predictions of proptein in 15 classes. An F1-score of 0.69 was attained by the Hybrid Xception model, which is superior to the F1-score of 0.61 attained by the conventional method. Summary of work done by various researchers for protein subcellular localization is given in Table 1.

Although methods have been developed to determine protein subcellular localization from microscopic images automatically, current systems still fall short of human performance. This study delves into the challenge of predicting protein distribution in human subcellular compartments. In this study, we classified HPA images into 28 different phenotypes based on the presence or absence of a specific protein. Hence, this paper gives a quick and effective approach for classifying human protein labels, with no requirement for laborious feature extraction and selection.

III. MATERIALS AND METHODS

This section starts with a detailed description of the dataset used for this study, followed by the pre-processing of the dataset before feeding the data to the proposed model. Then details about the proposed CNN model has been presented followed by the experimental setup required to train the model.

A. DATASET DESCRIPTION

Dataset utilized in his study has been acquired from a competition named ''Human Protein Atlas Image Classification''

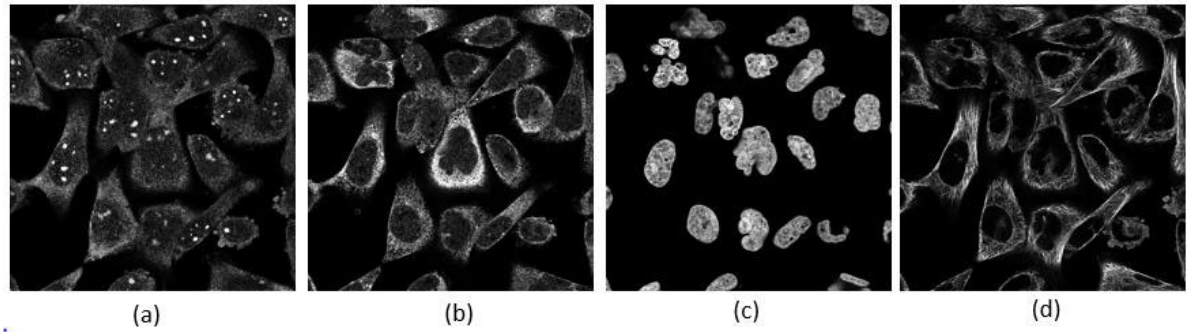

FIGURE 1. An example image with four colour channels: (a) green channel represents the protein of interest, (b) red channel represents microtubules, (c) blue channel represents nucleus, and (d) yellow channel representd endoplasmic reticulum.

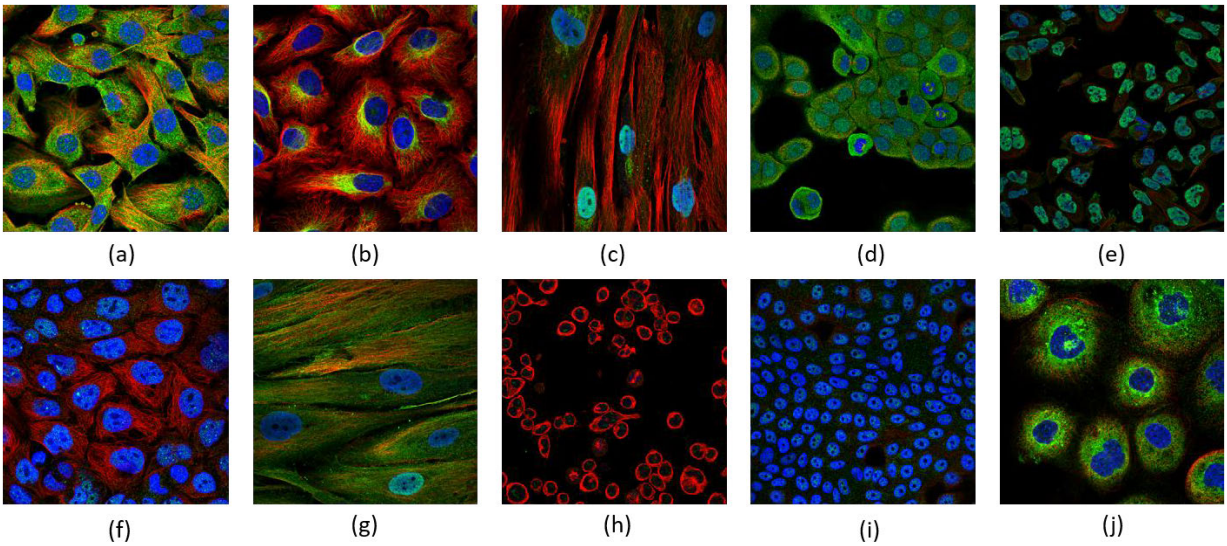

FIGURE 2. Sample images of multi-label or single-label from HPA dataset. Labels present in each image are: (a) Cytosol, nuclear membrane, plasma membrane, (b) Intermediate filaments, (c) Golgi apparatus, nucleoplasm, (d) Cytosol, plasma membrane, (e) Nucleoplasm, plasma membrane, (f) Nuclear bodies, (g) Actin filaments, cytosol, nucleoplasm, (h) Nucleoli, (i) Nucleoplasm, cytosol and (j) Endoplasmic reticulum.

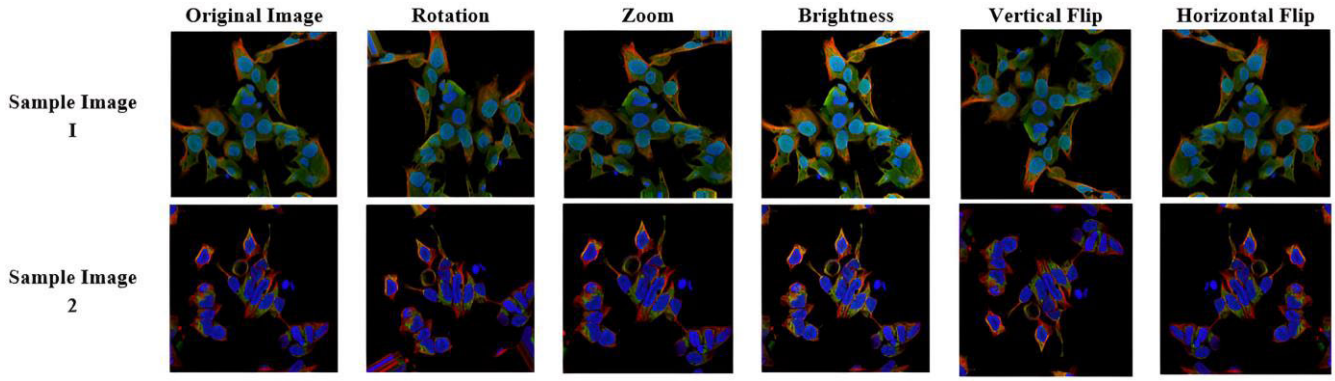

FIGURE 3. Sample augmented images after applying image transformations. Data augmentation has been done by applying transformations like random rotations, zoom, random brightness, vertical and horizontal flip.

held on Kaggle in 2019 [37]. This dataset contains 31,072 samples obtained using confocal microscopy.

The dataset, however, contains 27 different cell types with vastly diverse morphologies, which in turn affects the pattern

TABLE 2. Number fo samples of each class in training and test set.

Class No.	Class Name	Total	Train	Test
0	Nucleoplasm	12,885	10,306	2,579
1	Nuclear Membrane	1,254	999	255
2	Nucleoli	3,621	2,880	741
3	Nucleoli Fibrillar Centre	1,561	1,282	279
4	Nuclear Speckles	1,858	1,499	359
5	Nuclear Bodies	2,513	1,994	519
6	Endoplasmic Reticulum	1,008	811	197
7	Golgi Apparatus	2,822	2,251	571
8	Peroxisomes	53	42	11
9	Endosomes	45	36	9
10	Lysosomes	28	22	6
11	Intermediate Filaments	1,093	888	205
12	Actin Filaments	688	560	128
13	Focal Adhesion Sites	537	430	107
14	Microtubules	1,066	839	227
15	Microtubule End	21	17	4
16	Cytokinetic Bridge	530	419	111
17	Mitotic Spindle	210	165	45
18	Microtubule Organizing Centre	902	701	201
19	Centrosome	1,482	1,178	304
20	Lipid Droplets	172	143	29
21	Plasma Membrane	3,777	3,010	767
22	Cell Junctions	802	648	154
23	Mitochondria	2,965	2,358	607
24	Aggresome	322	255	67
25	Cytosol	8,228	6,560	1,668
26	Cytoplasmic Bodies	328	260	68
27	Rods and Rings	11	9	2

of proteins in the various organelles. All samples are four-channel confocal images.

The immunofluorescence-labelled protein of interest is shown to be localized with the green channel in the corresponding image. Three other reference channels are also been given, like, the blue channel, which displays nuclei counterstained with DAPI; the red channel, which displays microtubules labelled with an antibody against tubulin; and the yellow channel, which displays endoplasmic reticulum (ER) [11].

The four channels of an image sample are shown in Figure 1. For this study, out of the four channels available, only three channels were used to make an RGB image The distribution of proteins is categorized into 28 primary cellular organelles. The name of classes along with an assigned class number are given in Table 2. The number of image samples belonging to each class is shown in Table 2. It can be seen from this table that there is a huge data imbalance in the dataset. For class 0 there are as high as 12,885 image samples, while for class 27 there are as low as 27 image samples. Such a huge data imbalance hampers the performance of the model, as the model gets biased more towards the majority classes. Another issue due to data imbalance occurs when data splitting is done in train and test. The remedy for this has been discussed under data pre-processing in the subsequent section.

Another important aspect of the dataset to keep in mind for this study is that it is a multi-label classification problem. Multilabel means that each sample image may contain the protein location in more than one subcellular compartment. Figure 2 shows the sample RGB images obtained after concatenation of blue, green and red filters for every sample given in the dataset having multiple as well as single labels.

B. DATASET PRE-PROCESSING

The dataset was provided with JPEG images with a high-resolution size of 512 × 512. To check the effect of image scaling on the proposed convolutional neural network, the image size was resized to 128 × 128 and 256 × 256 as well. For faster training of the model, images were normalized first by dividing each pixel by 255. The pre-processed samples were then randomly split into a training (80%) and a test (20%) set. For an imbalanced dataset, it is recommended to use iterative stratification for data splitting.

Iterative stratification guarantees that the percentage of samples from each target class remains roughly the same in both the train and test sets. Data has been split using iterative stratification into the train (80%) and test (20%) data as shown in Table 2. The distribution of samples for each class in the training and test set is presented in Table 2.

C. DATA AUGMENTATION

To increase the size of the dataset, data augmentation is done by applying random transformation on the train images. It becomes necessary to add more images corresponding to minority classes in order to make model learn effectively for unbiased predictions because the dataset used is highly imbalanced. Increase in size of dataset, doesn't mean adding more samples to the dataset, perhaps applying random transformations on the images to make the proposed model generalizable. The following transformations have been applied on the dataset: random rotation in the range 0 to 270 degrees, zoom of 0.2, random brightness in range of 0 to 0.5, vertical flip and horizontal flip. Transformations applied on sample images are shown in Figure 3.

D. PROPOSED CNN MODEL

The use of CNN based architectures has been done to help advance biotechnology. For blood cancer identification, for instance, Liang et al. [38] combined a deep convolutional neural network with a recurrent neural network (RNN).

To classify images of HEp-2 cells, Gao et al. [39] created a model based on CNN and obtained the desired generalizability across datasets. In this paper, we present a convolutional neural network that can handle high-resolution images and, as a result, can learn from fine visual structures without having to downscale them.

A two-step process has accomplished this. In the initial phase, a four-block convolution encoder was utilized. In each convolutional block there are 2 convolutional layers and a max-pooling layer to discover abstract features at various spatial resolutions. The number of filters in each convolution block has been increased, such that the first convolution block generates 32 feature maps, the second generates 64 feature

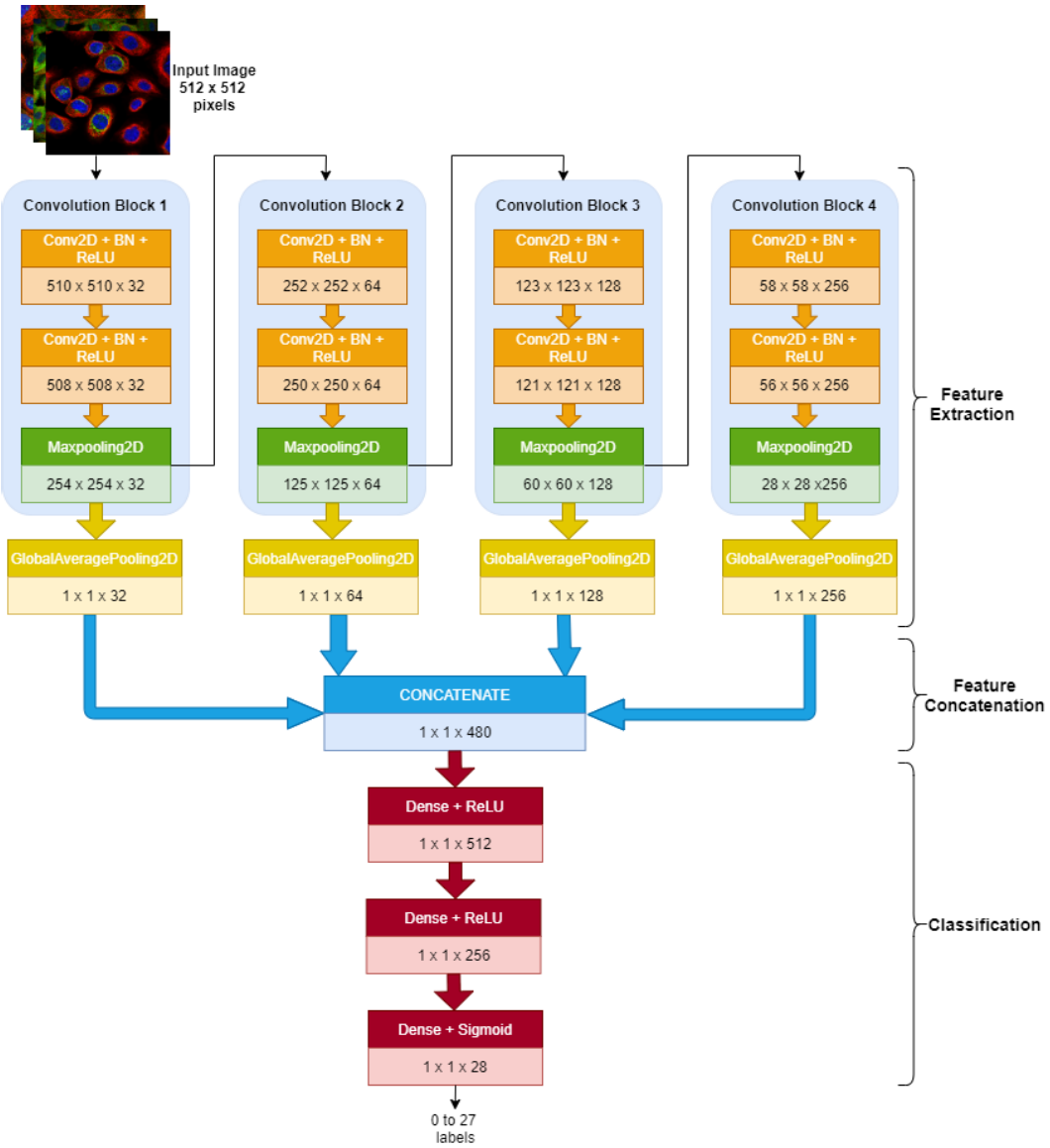

FIGURE 4. Architecture of proposed CNN model.

maps, and so on. The architecture of the proposed CNN model is given in Figure 5.

In the second stage, the feature maps created from the max pooling layer of each convolution block were shrunk to a size of one pixel using global average pooling. Global average pooling provides extreme dimension reduction and aids in preventing model overfitting. This pooling procedure can also reduce the impact of weak labels. The features generated from average global pooling layers were concatenated to form a feature vector. The resultant feature vector, which represents various spatial resolutions, is then fed to a fully connected network comprising of 2 dense layers for the final prediction. The first hidden layer consists of 512 neurons followed by the second hidden layer with 256 neurons. The output layer consists of 28 neurons, each neuron representing one output class. Since in multilabel classification, each label is treated as a binary classifier, therefore sigmoid function has been used to obtain the probabilities corresponding to each label. Additionally, batch normalization was carried out between each convolutional and activation layer and the Dense and Activation layer. The performance of the neural network can be enhanced by using this method, which involves normalising the inputs at each layer so that the output activation has a mean of 0 and a standard deviation of 1. The model summary of the proposed CNN architecture is given in Table 3.

E. EXPERIMENTAL SETUP

The specifics of the training process, in addition to the network architecture, are crucial for comprehending implementation and evaluating performance of the mofdel. Here, we outline the parameter selections made when the proposed CNN model was being trained. 27,965 samples were used to

TABLE 3. Model summary of proposed CNN.

Layer Type	Input Tensor	Filter Size	Activation Function	Output Tensor	No. of parameters
Input	512 x 512 x 3	----	----	-----	----
Conv2D	512 x 512 x 3	3, 3	----	510 x 510 x 32	896
Batch Normalization (BN)	510 x 510 x 32	----	----	----	128
Activation	510 x 510 x 32	----	ReLU	----	0
Conv2D	510 x 510 x 32	3, 3	----	508 x 508 x 32	9248
Batch Normalization	508 x 508 x 32	----	----	----	128
Activation	508 x 508 x 32	----	ReLU	----	0
Maxpooling2D	508 x 508 x 32	2, 2	----	254 x 254 x 32	0
Conv2D	254 x 254x 32	3, 3	----	252 x 252 x 64	18496
Batch Normalization	252 x 252 x 64	----	----	----	256
Activation	252 x 252 x 64	----	ReLU	----	0
Conv2D	252 x 252 x 64	3, 3	----	250 x 250 x 64	36928
Batch Normalization	250 x 250 x 64	----	----	----	256
Activation	250 x 250 x 64	----	ReLU	----	0
Maxpool2D	250 x 250 x 64	2, 2	----	125 x 125 x 64	0
Conv2D	125 x 125 x 64	3, 3	----	123 x 123 x 128	73856
Batch Normalization	123 x123 x128	----	----	----	512
Activation	123 x123 x128		ReLU	----	0
Conv2D	123 x123 x128	3, 3	----	121 x121 x128	147584
BatchNormalization	121 x121 x128	----	----	----	512
Activation	121 x121 x128	----	ReLU	----	0
Maxpool2D	121 x121 x128	2, 2	----	60 x 60 x 128	0
Conv2D	60 x 60 x 128	3, 3	----	58 x 58 x 256	295168
BatchNormalization	58 x 58 x 256	----	----	----	1024
Activation	58 x 58 x 256	----	ReLU	----	0
Conv2D	58 x 58 x 256	3, 3	----	56 x 56 x 256	590080
BatchNormalization	56 x 56 x 256	----	----	----	1024
Activation	56 x 56 x 256	----	ReLU	----	0
MaxPool2D	56 x 56 x 256	2, 2	----	28 x 28 x 256	0
GlobalAveragePooling2D	254 x 254 x 32	----	----	32 units	0
GlobalAveragePooling2D	125 x 125 x 64	----	----	64 units	0
GlobalAveragePooling2D	60 x 60 x 128	----	----	128 units	0
GlobalAveragePooling2D	28 x 28 x 256	----	----	256 units	0
Concatenate	32 units 64units 128 units 256 units	----	----	480 units	0
Dense	480 units	----	----	512 units	2,45,760
BatchNormalization	512 units	----	----	----	2.048
Activation	512 units	----	ReLU	----	0
Dense	512 units	----	----	256 units	1,31,072
BatchNormalization	256 units	----	----	----	4,096
Activation	256 units	----	ReLU	----	0
Dense	256 units	----	Sigmoid	28 units	7,168

train the model. Without utilizing transfer learning from previously trained models, the network was trained from scratch. The training hyperparameters were adjusted by experimentation. The model was tested with different optimizers like Adagrad, Adam, SGD and Adam. Different batch sizes of 8, 6 and 32were used to test the performance of the model. Maximum batch size that could be taken was 32 because of the out of memory allocation error by 12 GB GPU. The following options were ultimately selected for use in training the model. Glorot initialization was used to set the model weights initially. To train the models, Adam optimizer was used. The default parmeters of original Adam paper were used. 0.001 was set as the initial learning rate. When there was no decrease in validation loss, the learning rate was reduced by 0.1 after a patience of 3 epochs. Due to the presence of many labels in the data, the binary cross-entropy was selected as the loss function. The batch size of 32 employed in the batch normalization process was the maximum that could be accommodated by a graphics processing unit (GPU) with 12 GB of memory. The model was written in Python using the Keras module and the Tensorflow backend, and training was expedited utilising graphics processing units. CNN's confidence values are the model's raw outputs. The final layer's sigmoid activation function ensures that the output is always a positive number. For each class, the optimal confidence threshold was determined empirically by testing various values and seeing which one yielded the highest F1-score on the training data. The threshold values were tested at a resolution of 0.01. The hyperparameter settings for the proposed model are shown in Table 4.

IV. RESULTS AND DISCUSSIONS

The metrics used to evaluate the performance of the model and discussions about results obtained with the proposed

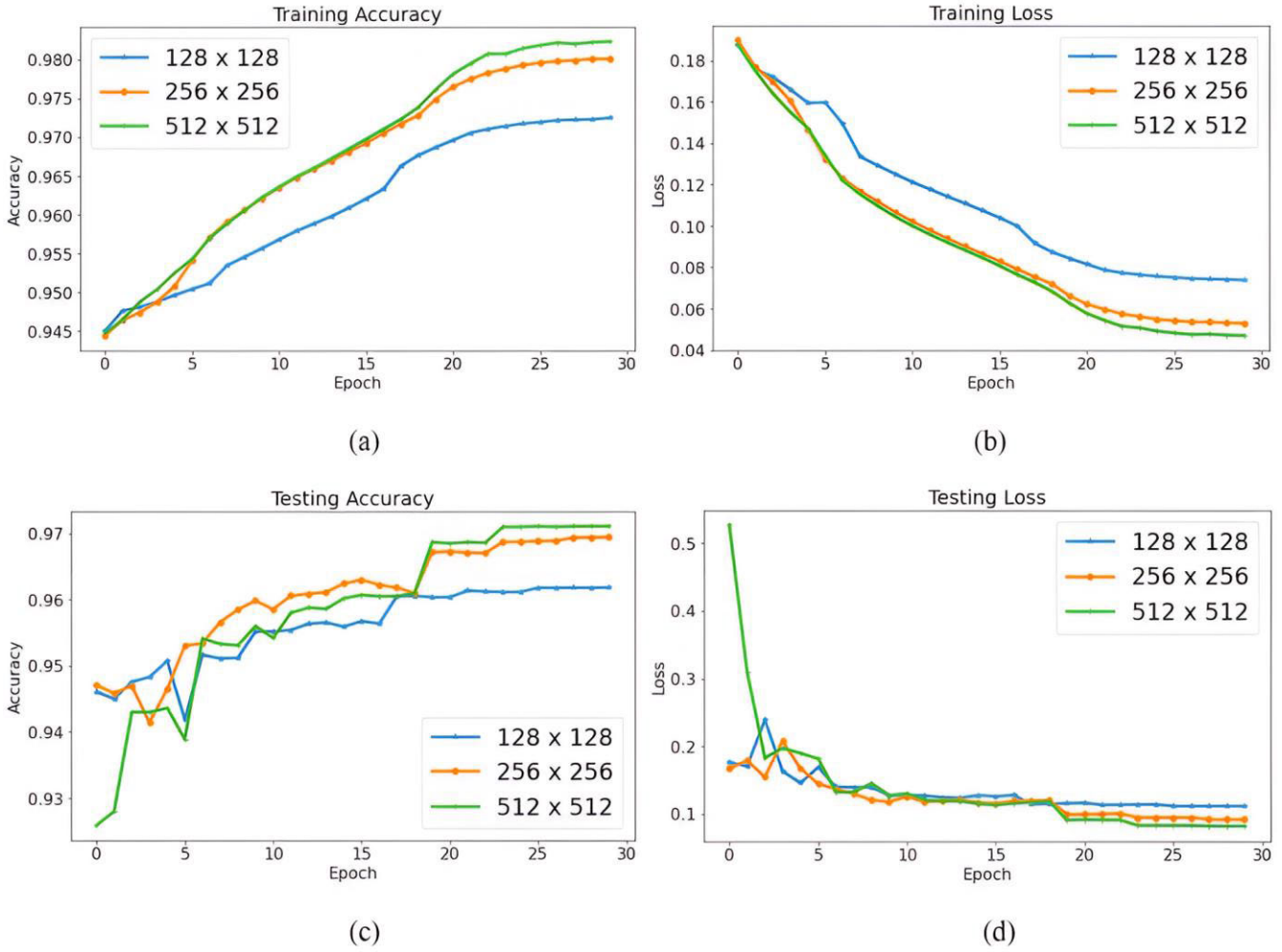

FIGURE 5. Accuracy and loss plots obtained during training of the proposed CNN model when applied with different input image sizes (a) Training accuracy, (b) Training loss, (c) Testing accuracy and (d) Testing loss. Maximum accuracy and minimum loss have been obtained when the network was fed with image size of 512 × 512.

CNN model on various configurations is presented in this section.

A. PERFORMANCE METRICS

The model was evaluated on three main performance metrics: recall, precision and F1-score. These are calculated using 4 parameters namely, TP (True Positive), FP (False Positive), TN (True negative) and FN (False Negative).

1) PRECISION

Precision is defined as the out of all predicted positives, how many samples are true positives. Its formula is given by equation 1.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

2) RECALL

Recall defines the true positives out of actual positives. Its formula is given by equation 2.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

TABLE 4. Hyperparameter configurations.

Hyperparameters	Values
Mini Batch size	32
Initial Learning Rate	0.001
Weight Decay	10^{-8}
Beta	0.9, 0.999
Adam	True

3) F1-SCORE

Precision and recall are averaged to form the F1-score. and its formula is given by Equation 3.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

Due to the extreme asymmetry of the training data utilised in this study, the sample sizes assigned to each label must be taken into account when calculating average recall, precision and F1-score. As a result, as shown in Figures 6 and 8, we computed a weighted average of the recall, precision and F1-score. Additionally, though the model's performance has mainly been evaluated using F1-score, recall and precision, there are some other performance criteria too which can be calculated for multilabel image classification.

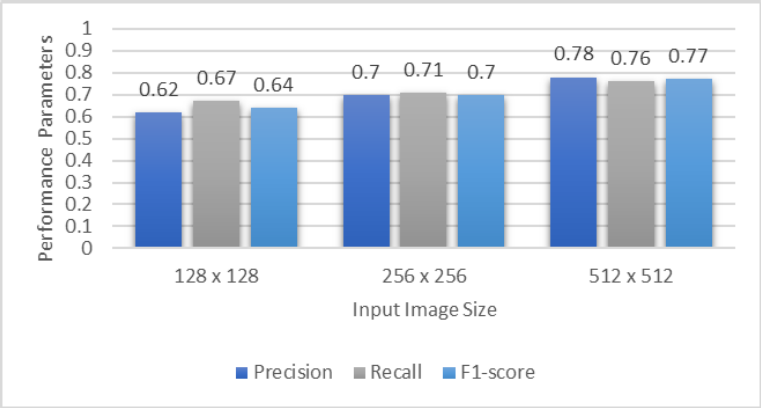

FIGURE 6. Performance of the model based on the basis of different input shapes. Maximum recall, precision and f1-score has been obtained when the model was fed with input image size of 512 × 512.

TABLE 5. Performance metrics of each label on different input shape.

Input shape	128 X 128			256 X 256			512 X 512		
Label No.	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0.77	0.87	0.82	0.80	0.88	0.84	0.86	0.90	0.88
1	0.79	0.75	0.77	0.84	0.82	0.83	0.90	0.87	0.89
2	0.68	0.63	0.65	0.82	0.74	0.78	0.88	0.77	0.82
3	0.50	0.54	0.52	0.70	0.69	0.69	0.80	0.71	0.75
4	0.71	0.69	0.70	0.82	0.78	0.80	0.88	0.80	0.85
5	0.46	0.48	0.47	0.69	0.63	0.66	0.71	0.67	0.69
6	0.43	0.43	0.43	0.62	0.48	0.54	0.56	0.57	0.57
7	0.58	0.58	0.58	0.77	0.70	0.73	0.82	0.73	0.77
8	0.11	0.36	0.16	0.47	0.73	0.57	0.62	0.73	0.67
9	0.27	0.33	0.30	0.70	0.78	0.74	0.58	0.78	0.67
10	0.67	0.67	0.67	0.67	0.67	0.67	0.60	1.00	0.75
11	0.57	0.44	0.50	0.83	0.58	0.68	0.74	0.71	0.72
12	0.37	0.33	0.35	0.67	0.48	0.56	0.64	0.58	0.61
13	0.51	0.36	0.42	0.69	0.49	0.57	0.85	0.47	0.60
14	0.88	0.79	0.83	0.88	0.81	0.84	0.87	0.86	0.86
15	1.00	0.25	0.40	1.00	0.50	0.67	1.00	0.50	0.67
16	0.09	0.37	0.15	0.26	0.31	0.28	0.36	0.26	0.30
17	0.14	0.29	0.19	0.23	0.31	0.26	0.39	0.24	0.30
18	0.34	0.39	0.36	0.41	0.50	0.45	0.51	0.41	0.46
19	0.29	0.36	0.32	0.58	0.57	0.57	0.66	0.60	0.63
20	0.25	0.24	0.25	0.41	0.45	0.43	0.48	0.41	0.44
21	0.55	0.68	0.61	0.64	0.69	0.66	0.67	0.68	0.68
22	0.30	0.43	0.35	0.52	0.51	0.52	0.56	0.48	0.52
23	0.75	0.59	0.66	0.85	0.72	0.78	0.78	0.80	0.79
24	0.74	0.34	0.47	0.73	0.60	0.66	0.73	0.61	0.67
25	0.60	0.76	0.67	0.65	0.74	0.69	0.63	0.81	0.71
26	0.15	0.26	0.19	0.46	0.44	0.45	0.43	0.47	0.45
27	0.50	0.50	0.50	1.00	0.50	0.67	1.00	0.50	0.67

4) SPECIFICITY

Specificity defines how accurately has the model predicted true negatives. The formula gives it in quation 4.

$$Specificity = \frac{TN}{FP + TN} \tag{4}$$

5) FALSE POSITIVE RATE (FPR)

False Positive Rate returns the percentage of true negative samples which were predicted as false positives. FPR can be calculated using equation 5.

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

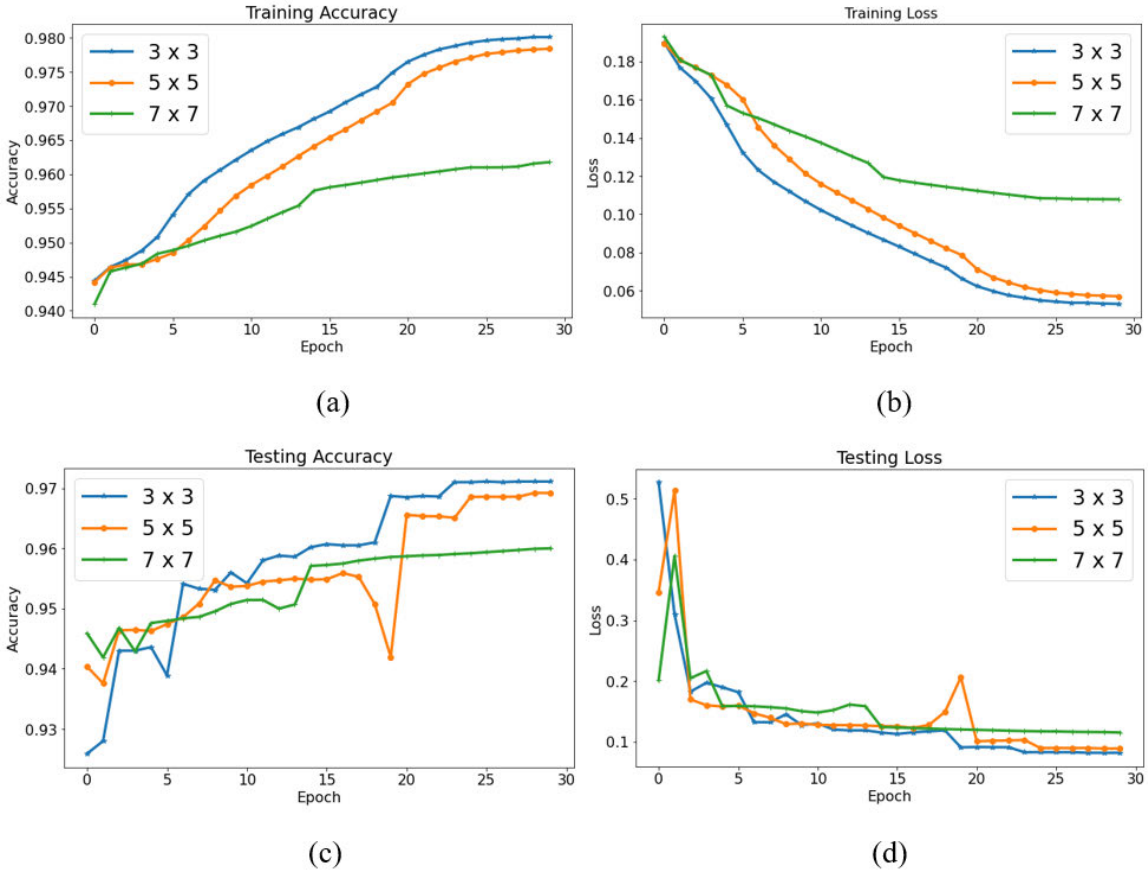

FIGURE 7. Accuracy and loss plots for different filter size (a) Training accuracy, (b) Training loss, (c) Testing accuracy and (d) Testing loss. Maximum accuracy and minimum loss have been achieved when filter size of 3 × 3 was used in convolutional layers of the proposed model.

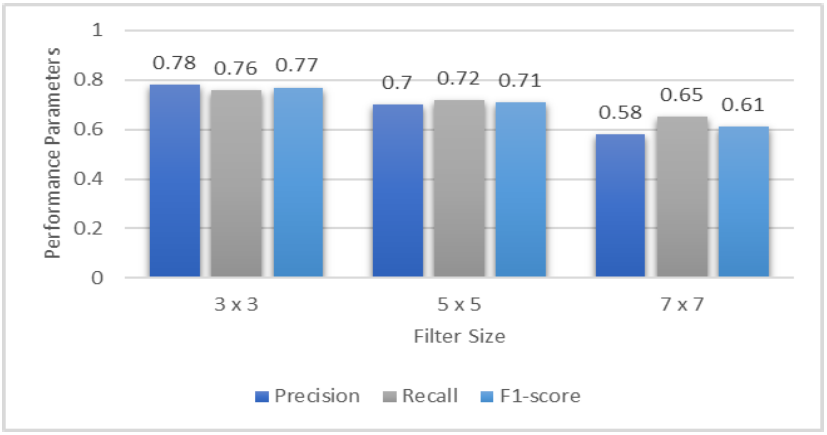

FIGURE 8. Performance of the model based on different Filter Size. Maximum recall, precision and f1-score have been achieved when filter size of 3 × 3 was used in convolutional layers of the proposed model.

6) FALSE NEGATIVE RATE (FNR)

False negative returns the percentage of true positive samples which were predicted as false negatives. The formula to calculate FNR is given in equation 6.

$$FNR = \frac{FN}{TP + FN} \tag{6}$$

7) AUC-ROC SCORE

AUC-ROC curve is another useful indicator for assessing a model's efficacy. The model's efficacy at varying thresholds is depicted graphically by the plot between sensitivity and False Positive Rate. Threshold values were taken in the range of 0 to 1 at interval of 0.01. A higher AUC indicates better model performance.

TABLE 6. Performance metrics of each label on different filter size.

Filter Size	3 X 3			5 X 5			7 X 7		
Label No.	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
0	0.86	0.90	0.88	0.80	0.89	0.84	0.78	0.85	0.81
1	0.90	0.87	0.89	0.89	0.78	0.83	0.70	0.55	0.62
2	0.88	0.77	0.82	0.83	0.72	0.77	0.60	0.61	0.61
3	0.80	0.71	0.75	0.59	0.64	0.61	0.50	0.41	0.45
4	0.88	0.80	0.85	0.80	0.79	0.80	0.74	0.74	0.74
5	0.71	0.67	0.69	0.68	0.58	0.63	0.30	0.51	0.38
6	0.56	0.57	0.57	0.51	0.55	0.53	0.55	0.56	0.55
7	0.82	0.73	0.77	0.83	0.60	0.69	0.55	0.56	0.55
8	0.62	0.73	0.67	0.38	0.73	0.50	0.43	0.27	0.33
9	0.58	0.78	0.67	1.00	0.56	0.71	0.10	0.56	0.17
10	0.60	1.00	0.75	0.71	0.83	0.77	0.14	0.50	0.22
11	0.74	0.71	0.72	0.73	0.56	0.64	0.51	0.36	0.42
12	0.64	0.58	0.61	0.48	0.52	0.50	0.37	0.36	0.36
13	0.85	0.47	0.60	0.75	0.42	0.54	0.41	0.38	0.40
14	0.87	0.86	0.86	0.89	0.80	0.84	0.86	0.70	0.77
15	1.00	0.50	0.67	0.50	0.25	0.33	0.02	0.25	0.03
16	0.36	0.26	0.30	0.26	0.15	0.19	0.15	0.14	0.14
17	0.39	0.24	0.30	0.21	0.24	0.23	0.11	0.20	0.14
18	0.51	0.41	0.46	0.33	0.54	0.41	0.36	0.29	0.32
19	0.66	0.60	0.63	0.45	0.48	0.46	0.23	0.42	0.29
20	0.48	0.41	0.44	0.24	0.31	0.27	0.38	0.21	0.27
21	0.67	0.68	0.68	0.65	0.65	0.65	0.51	0.59	0.55
22	0.56	0.48	0.52	0.50	0.46	0.48	0.35	0.42	0.38
23	0.78	0.80	0.79	0.75	0.75	0.75	0.64	0.64	0.64
24	0.73	0.61	0.67	0.53	0.61	0.57	0.74	0.42	0.53
25	0.63	0.81	0.71	0.64	0.77	0.70	0.56	0.78	0.65
26	0.43	0.47	0.45	0.38	0.46	0.42	0.17	0.16	0.17
27	1.00	0.50	0.67	1.00	0.50	0.67	1.00	0.50	0.67

B. EVALUATION OF THE PROPOSED MODEL WITH DIFFERENT INPUT IMAGE SIZES

To gauge the efficacy of the model, a series of tests were run using input images of varying sizes (512 × 512 pixels, 256 × 256 pixels, and 128 × 128). The results of these studies are shown in Figure 5 in the form of loss and accuracy plots. With a 512-by-512-pixel input image, a maximum training accuracy of 97.72% was achieved as shown in figure 5(a) and maximum test accuracy of 97.03% was achieved as shown in Figure 5(c). Training and testing losses for the 512 × 512 input shape dropped to 0.061 and 0.084, respectively, as illustrated in Figures 5(b) and 5(d). Table 5 and Figure 6 show each label's average values of recall, F1-score and precision on different input image sizes, respectively. From Figure 6, we can deduce that a 0.74 F1-score was achieved with an input image size of 512 pixels by 512 pixels.

C. EVALTUATION OF THE PROPOSED MODEL WITH DIFFERENT FILTER SIZES

In order to demonstrate the efficacy of the proposed CNN model and to examine the model's performance on microscopic images, experiments were also conducted with varying filter sizes. For these tests, 512-by-512-pixel images were used. Three different sizes of filter were used to measure the model's effectiveness The chosen filter sizes were: 3 × 3, 5 × 5 and 7 × 7. Plots of loss and accuracy on different filter sizes are shown in Figure 7. Based on the results shown in Figure 7, it is evident that 3 × 3 filter size yielded the best results across the board. The effectiveness of the model degrades as the filter size grows larger. Comparison of the average recall, precision and F1-score for three distinct filter sizes is shown in Figure 8. Table 6 provides the performance metrics achieved using different filter sizes for each label.

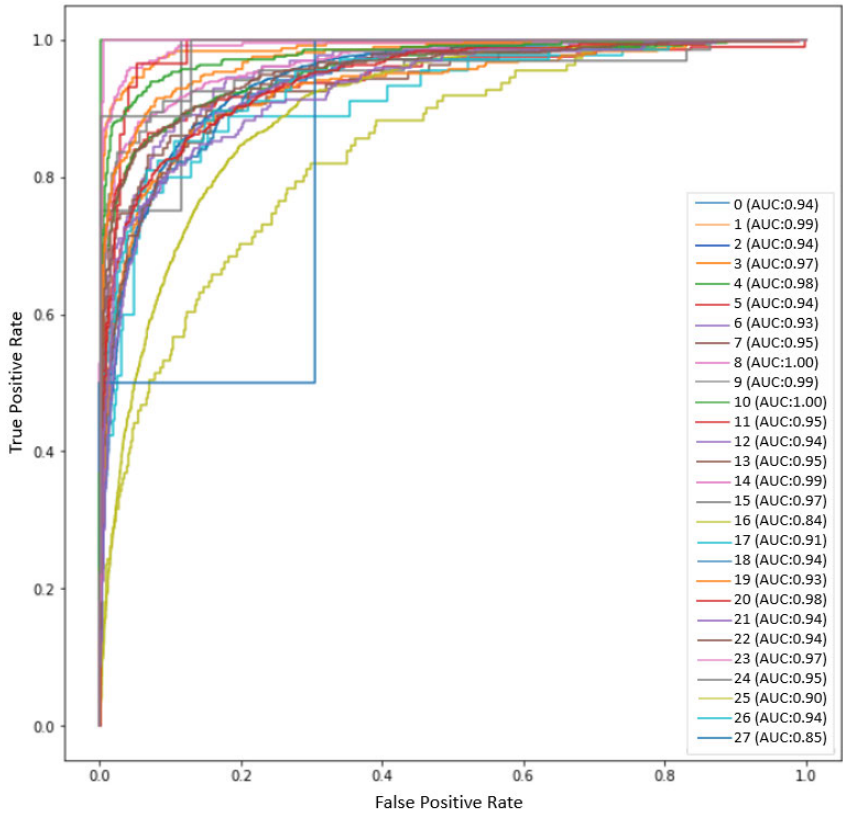

FIGURE 9. ROC Curve for the proposed CNN model.

This shows that the proposed model was able to classify true negatives at an excellent rate. Similarly, False Positive Rate is also very low which implies that there were very rare true negatives that were classified as true positives. In other words, we can say that the absence of protein in a subcellular compartment has been predicted with high accuracy. In comparison to that, there are some labels for which the false negative rate is somewhat on the higher side. Like from Table 7 it can be seen that the cytokinetic bridge and mitotic spindle shows a very high false negative rate. This implies that the presence of proteins in these compartments has not been recognized by the model very effectively. Figure 9 shows the ROC curve for each label. AUC score of each label is given in Table 7. The higher the AUC score better is the performance of the model. Table 6 shows that the AUC score of almost all the labels is excellent. This shows our model can efficiently locate the proteins in the correct subcellular compartments.

D. VISUALIZATION OF PREDICTION RESULTS

Figure 10 shows the examples of correct predictions done by the proposed model. In the left column of the image are the sample images with true labels and the prediction probabilities obtained for each layer from the output layer are shown in the right column. The output layer of the proposed CNN model consists of a sigmoid activation function as each label has to be assigned 0 or 1 according to the predicted probability value above or below the threshold. The model performed best on the threshold value of 0.5. That means, if the probability for a label is greater than 0.5, then it will be assigned 1 otherwise label will be 0 for that particular sample. As shown in Figure 10(a), the true labels of the image sample are 0 and 23, and the corresponding probability graph also shows that the predicted labels were also 0 and 23. It can be observed in Figure 10, that the labels with the majority of the image samples have been predicted mostly correctly. Like labels 0 and 25 have been correctly classified in all of the images.

Similarly, Figure 11 shows examples of incorrect classification. Again, the majority of classes have been predicted by the model almost accurately in all the cases i.e., labels 0 and 25. There are some cases like Figure 11(b) where even though there is no protein in the cytosol (label 25) in the image sample, but still, the model has predicted the presence of protein in 25. This shows that the major impact on the performance of the model can be due to data imbalance.

E. COMPARISON WITH THE STATE-OF-ART

To validate the results obtained using proposed CNN model, state-of-art comparison has been made in Table 8. We have chosen F1-score as the performance parameter to be compared as it takes into account both precision and recall. It can be seen from the table, that the proposed model in

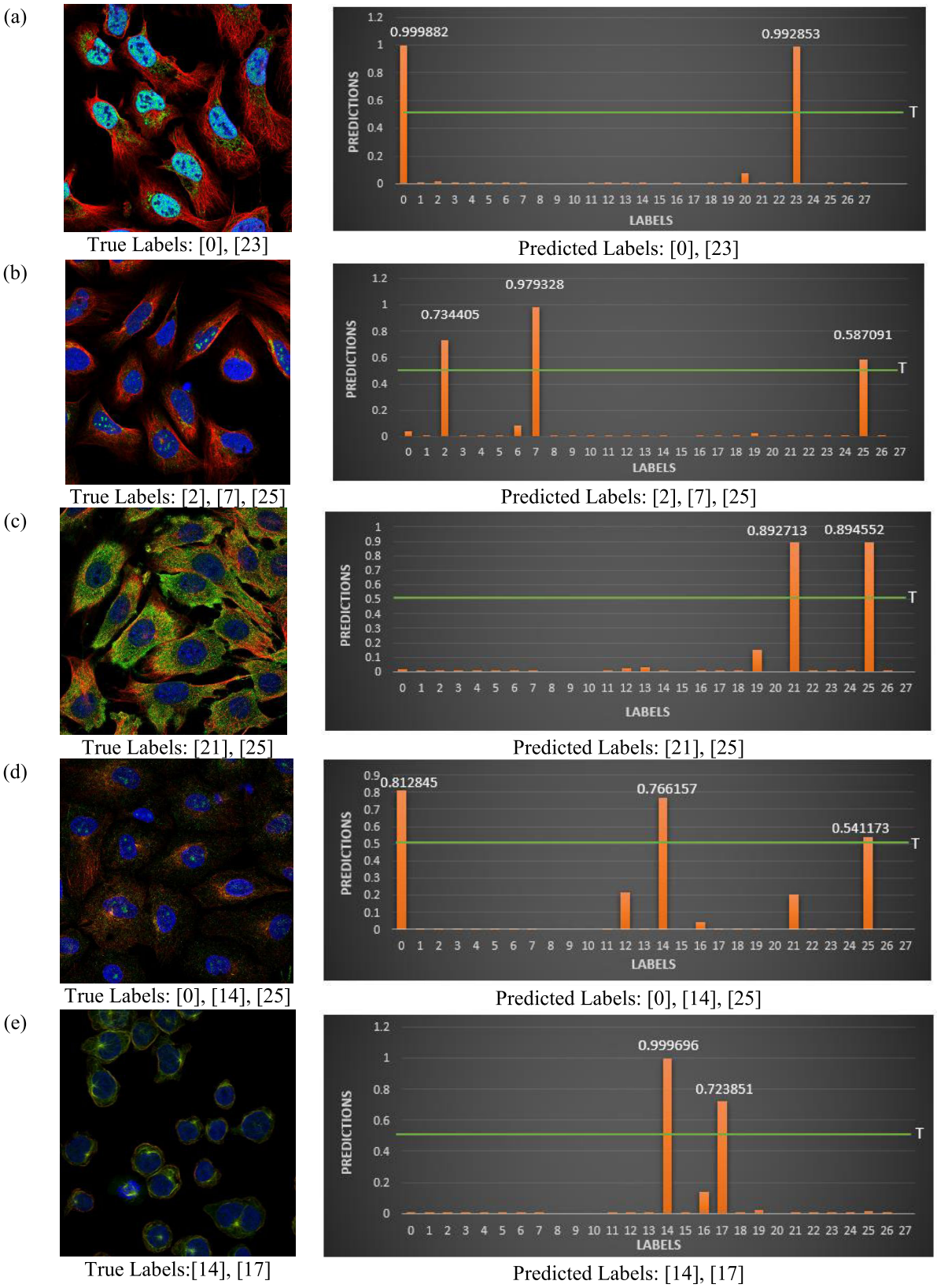

FIGURE 10. Examples of correct classification by the proposed CNN Model. Left column represents the original image with true labels and right column represents the predicted probabilities obtained from the model corresponding to the original image.

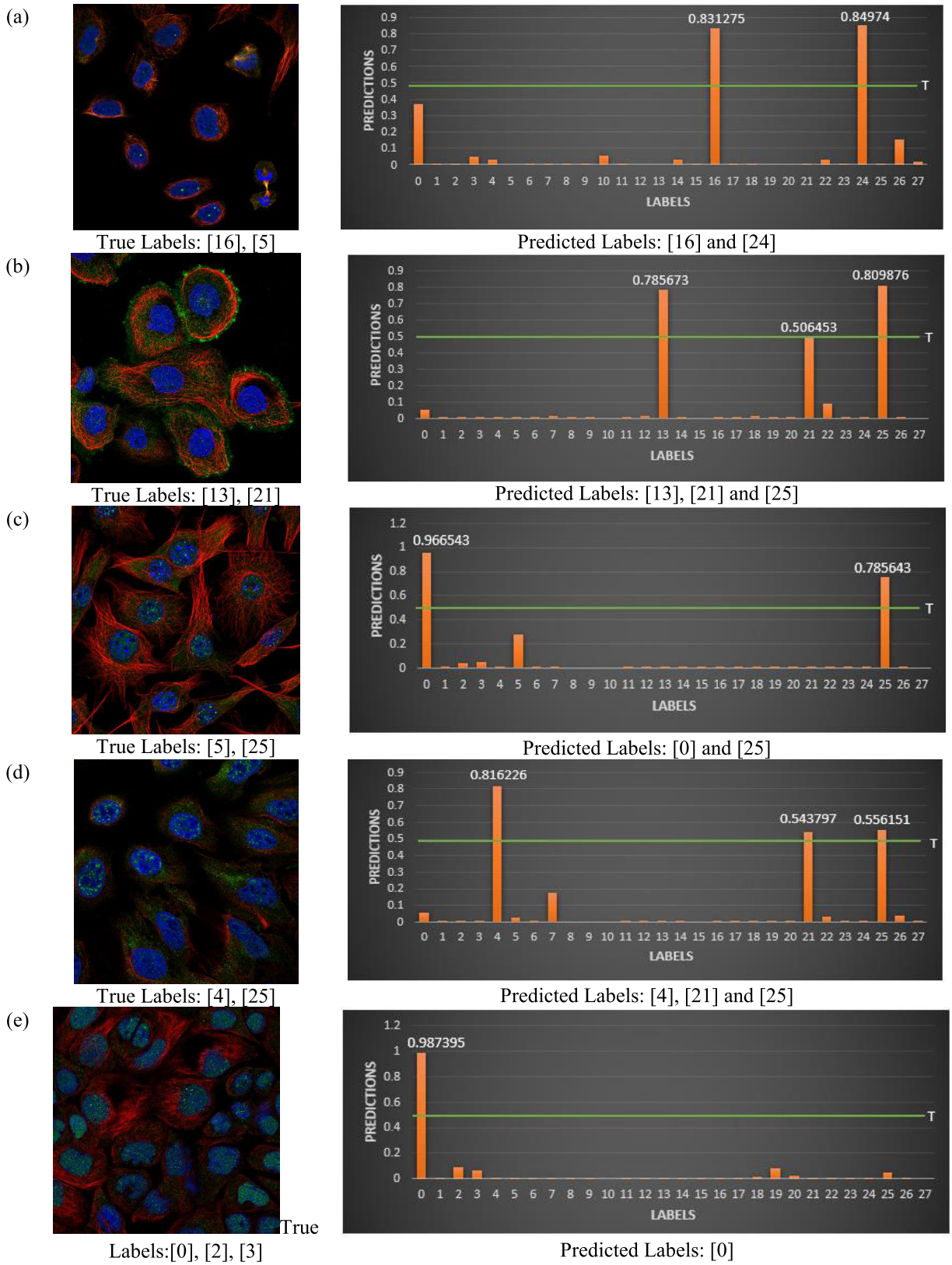

FIGURE 11. Examples of incorrect classification by the proposed CNN model. Left column represents the original image with true labels and right column represents the predicted probabilities obtained from the model corresponding to the original image.

TABLE 7. Performance of the proposed model in terms of specificity, FPR, FNR and AUC score.

Label No.	Label Name	Specificity	False Positive Rate (FPR)	False Negative Rate (FNR)	AUC Score
0	Nucleoplasm	0.87	0.13	0.12	0.94
1	Nuclear Membrane	0.99	0.01	0.19	0.99
2	Nucleoli	0.97	0.03	0.22	0.94
3	Nucleoli Fibrillar Centre	0.99	0.01	0.29	0.97
4	Nuclear Speckles	0.99	0.01	0.21	0.98
5	Nuclear Bodies	0.98	0.02	0.34	0.94
6	Endoplasmic Reticulum	0.98	0.02	0.32	0.93
7	Golgi Apparatus	0.98	0.02	0.25	0.95
8	Peroxisomes	0.99	0.01	0.27	1.00
9	Endosomes	1.00	0.00	0.33	0.99
10	Lysosomes	1.00	0.00	0.17	1.00
11	Intermediate Filaments	0.99	0.01	0.30	0.95
12	Actin Filaments	0.99	0.01	0.40	0.94
13	Focal Adhesion Sites	1.00	0.00	0.50	0.95
14	Microtubules	1.00	0.00	0.18	0.99
15	Microtubule End	1.00	0.00	0.50	0.97
16	Cytokinetic Bridge	0.99	0.01	0.77	0.84
17	Mitotic Spindle	0.99	0.01	0.67	0.91
18	Microtubule Organizing Centre	0.98	0.02	0.41	0.94
19	Centrosome	0.98	0.02	0.42	0.93
20	Lipid Droplets	0.99	0.01	0.55	0.98
21	Plasma Membrane	0.94	0.06	0.26	0.94
22	Cell Junctions	0.99	0.01	0.49	0.94
23	Mitochondria	0.98	0.02	0.19	0.97
24	Aggresome	1.00	0.00	0.39	0.95
25	Cytosol	0.85	0.15	0.22	0.90
26	Cytoplasmic Bodies	1.00	0.00	0.56	0.94
27	Rods and Rings	1.00	0.00	0.50	0.85

TABLE 8. Comparison of the proposed CNN model with the sate-of-art.

Reference No.	Technique	Dataset Used	F1-score
[34]	FCN	HPA Dataset	0.696
	CNN		0.676
[35]	Inception V3	HPA Dataset	0.706
[36]	Hybrid Xception	HPA Dataset	0.69
[11]	CNN based tool Loc-CAT	HPA Dataset	0.72
[38]	DeepLoc	Yeast FGP Dataset	0.72
[39]	ResNet	HPA Dataset	0.3459
[40]	CNN	HPA Datset	0.74
[31]	CNN	HPA Dataset	0.83
Proposed Model	CNN	HPA Dataset	**0.77**

this research outperformed the other models by obtaining an F1-score of 0.77. It can be seen from Table 8 that in [31], F1-score obtained is 0.83 while the proposed model achieved an F1-score of 0.77 for protein subcellular localization in 28 subcellular compartments. The reason for better result obtained in [31] is that they have used a greater number of samples i.e., 50,864 images. With a greater number of images, class imbalance is low, leading to better results. Also, in this paper the precision, recall and F1-score of individual labels has not been given. Since our dataset is highly imbalanced, if we calculate the average precision and recall of top-5 best predicted classes it comes out to be 0.93, 0.88 and 0.86 respectively.

V. CONCLUSION AND FUTURE SCOPE

We have shown that the proposed CNN model may be used to automatically label protein positions in immunostained high throughput imaging data, and we anticipate that it will eventually be used in routine clinical applications. It is a general-purpose, dependable method for learning images from a large range of heterogeneous cell lines, and it can handle images of any size as input. While its present predictive performance on a diverse dataset is 0.77, it has space to increase with the inclusion of data from fresh stains and cell lines.

Our findings demonstrate that the class imbalance significantly impacted the outcomes. It is well-known that deep learning-based algorithms perform well when there is

abundant data, therefore it is reasonable to assume that the low amount of samples in some classes makes the model more prone to misclassifications.

Like, despite their different locations, actin filaments are sometimes mislabeled as the nucleus. It can be seen from the results that misclassifications into the most common classes in the training data—cytosol and nucleoplasm were very common. So, the main limitation of this research work is the highly imbalanced nature of dataset. In machine learning, the class imbalance is prevalent, but it may be fixed in several ways, including over- and under-sampling or creating fresh data for the rare classes. But this is not so simple in the case of a multi-label problem, because rare classes frequently coexist with common classes, and so, oversampling minority classes would also oversample the majority classes. So future work entails the collection of more samples for the rare classes for more effective prediction.

REFERENCES

[1] L. Nanni, A. Lumini, and S. Brahnam, "Local binary patterns variants as texture descriptors for medical image analysis," *Artif. Intell. Med.*, vol. 49, no. 2, pp. 117–125, 2010.
[2] R. F. Murphy and V. M. B. M. Velliste, "Towards a systematics for protein subcellular location: Quantitative description of protein localization patterns and automated analysis of fluorescence microscope images," in *Proc. ISMB*, vol. 8, 2000, pp. 251–259.
[3] E. Glory and R. F. Murphy, "Automated subcellular location determination and high-throughput microscopy," *Develop. Cell*, vol. 12, no. 1, pp. 7–16, Jan. 2007.
[4] M. Tahir, A. Khan, and A. Majid, "Protein subcellular localization of fluorescence imagery using spatial and transform domain features," *Bioinformatics*, vol. 28, no. 1, pp. 91–97, Jan. 2012.
[5] C.-C. Lin, Y.-S. Tsai, Y.-S. Lin, T.-Y. Chiu, C.-C. Hsiung, M.-I. Lee, J. C. Simpson, and C.-N. Hsu, "Boosting multiclass learning with repeating codes and weak detectors for protein subcellular localization," *Bioinformatics*, vol. 23, no. 24, pp. 3374–3381, Dec. 2007.
[6] M. V. Boland and R. F. Murphy, "A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells," *Bioinformatics*, vol. 17, no. 12, pp. 1213–1223, Dec. 2001.
[7] G. Srinivasa, T. Merryman, A. Chebira, J. Kovacevic, and A. Mintos, "Adaptive multiresolution techniques for subcellular protein location classification," in *Proc. IEEE Int. Conf. Acoust. Speed Signal Process.*, vol. 5, May 2006, p. 5.
[8] A. Chebira, Y. Barbotin, C. Jackson, T. Merryman, G. Srinivasa, R. F. Murphy, and J. Kovačević, "A multiresolution approach to automated classification of protein subcellular location images," *BMC Bioinf.*, vol. 8, no. 1, pp. 1–10, Dec. 2007.
[9] R. F. Murphy, "Automated proteome-wide determination of subcellular location using high throughput microscopy," in *Proc. 5th IEEE Int. Symp. Biomed. Imag., From Nano Macro*, May 2008, pp. 308–311.
[10] M. Uhlen, P. Oksvold, L. Fagerberg, E. Lundberg, K. Jonasson, M. Forsberg, M. Zwahlen, C. Kampf, K. Wester, S. Hober, H. Wernerus, L. Björling, and F. Ponten, "Towards a knowledge-based human protein atlas," *Nature Biotechnol.*, vol. 28, no. 12, pp. 1248–1250, Dec. 2010.
[11] P. J. Thul et al., "A subcellular map of the human proteome," *Science*, vol. 356, no. 6340, 2017, Art. no. eaal3321.
[12] L.-F. Handfield, B. Strome, Y. T. Chong, and A. M. Moses, "Local statistics allow quantification of cell-to-cell variability from high-throughput microscope images," *Bioinformatics*, vol. 31, no. 6, pp. 940–947, Mar. 2015.
[13] M. V. Boland, M. K. Markey, and R. F. Murphy, "Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images," *Cytometry*, vol. 33, no. 3, pp. 366–375, Nov. 1998.
[14] C. Conrad, H. Erfle, P. Warnat, N. Daigle, T. Lörch, J. Ellenberg, R. Pepperkok, and R. Eils, "Automatic identification of subcellular phenotypes on human cell arrays," *Genome Res.*, vol. 14, no. 6, pp. 1130–1136, 2004.
[15] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digit. Signal Process.*, vol. 126, Jun. 2022, Art. no. 103514.
[16] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022.
[17] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, and R. Cucchiara, "From show to tell: A survey on deep learning-based image captioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 539–559, Jan. 2023.
[18] R. Davuluri and R. Rengaswamy, "Identification of Alzheimer's disease using various deep learning techniques—A review," in *Intelligent Manufacturing and Energy Sustainability*. Singapore: Springer, 2022, pp. 485–498.
[19] R. Zemouri, N. Zerhouni, and D. Racoceanu, "Deep learning in the biomedical applications: Recent and future status," *Appl. Sci.*, vol. 9, no. 8, p. 1526, Apr. 2019.
[20] S. Dash, B. R. Acharya, M. Mittal, A. Abraham, and A. Kelemen, Eds., *Deep Learning Techniques for Biomedical and Health Informatics*. Cham, Switzerland: Springer, 2020.
[21] Y. Park and M. Kellis, "Deep learning for regulatory genomics," *Nature Biotechnol.*, vol. 33, no. 8, pp. 825–826, 2015.
[22] L. Zhuang, J. Lipkova, R. Chen, and F. Mahmood, "Deep learning-based integration of histology, radiology, and genomics for improved survival prediction in glioma patients," *Proc. SPIE*, vol. 12039, Apr. 2022, Art. no. 120390Z.
[23] Y. Wang, G. Zhang, X. Liu, Y. Xi, P. Wang, Y. Zhang, and X. Li, "Abstract 5045: Genomics and pathology based deep learning to predict cancers of unknown primary," *Cancer Res.*, vol. 82, no. 12, p. 5045, Jun. 2022.
[24] J. M. Ede, "Deep learning in electron microscopy," *Mach. Learn., Sci. Technol.*, vol. 2, no. 1, Mar. 2021, Art. no. 011004.
[25] Z. Liu, L. Jin, J. Chen, Q. Fang, S. Ablameyko, Z. Yin, and Y. Xu, "A survey on applications of deep learning in microscopy image analysis," *Comput. Biol. Med.*, vol. 134, Jul. 2021, Art. no. 104523.
[26] X. Chen, M. Velliste, and R. F. Murphy, "Automated interpretation of subcellular patterns in fluorescence microscope images for location proteomics," *Cytometry A*, vol. 69A, no. 7, pp. 631–640, 2006.
[27] R. Su, L. He, T. Liu, X. Liu, and L. Wei, "Protein subcellular localization based on deep image features and criterion learning strategy," *Briefings Bioinf.*, vol. 22, no. 4, Dec. 2020, Art. no. bbaa313.
[28] Z. Liu, Z. Wang, and B. Du, "Multi-marginal contrastive learning for multilabel subcellular protein localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20626–20635.
[29] O. Z. Kraus, B. T. Grys, J. Ba, Y. Chong, B. J. Frey, C. Boone, and B. J. Andrews, "Automated analysis of high-content microscopy data with deep learning," *Mol. Syst. Biol.*, vol. 13, no. 4, p. 924, 2017.
[30] T. Pärnamaa and L. Parts, "Accurate classification of protein subcellular localization from high-throughput microscopy images using deep learning," *G3, Genes, Genomes, Genet.*, vol. 7, no. 5, pp. 1385–1392, May 2017.
[31] S. Xiang, Q. Liang, Y. Hu, P. Tang, G. Coppola, D. Zhang, and W. Sun, "AMC-Net: Asymmetric and multi-scale convolutional neural network for multi-label HPA classification," *Comput. Methods Programs Biomed.*, vol. 178, pp. 275–287, Sep. 2019.
[32] D. P. Sullivan, C. F. Winsnes, L. Åkesson, M. Hjelmare, M. Wiking, R. Schutten, L. Campbell, H. Leifsson, S. Rhodes, A. Nordgren, K. Smith, B. Revaz, B. Finnbogason, A. Szantner, and E. Lundberg, "Deep learning is combined with massive-scale citizen science to improve large-scale image classification," *Nature Biotechnol.*, vol. 36, no. 9, pp. 820–828, Oct. 2018.
[33] K. Liimatainen, M. Valkonen, L. Latonen, and P. Ruusuvuori, "Cell organelle classification with fully convolutional neural networks," Tech. Rep., 2018.
[34] K. Liimatainen, R. Huttunen, L. Latonen, and P. Ruusuvuori, "Convolutional neural network-based artificial intelligence for classification of protein localization patterns," *Biomolecules*, vol. 11, no. 2, pp. 1–15, 2021.
[35] Z. Li, R. Togo, T. Ogawa, and M. Haseyama, "Classification of subcellular protein patterns in human cells with transfer learning," in *Proc. IEEE 1st Global Conf. Life Sci. Technol. (LifeTech)*, Mar. 2019, pp. 273–274.
[36] T. R. Shwetha, S. A. Thomas, V. Kamath, and N. K. B., "Hybrid Xception model for human protein atlas image classification," in *Proc. IEEE 16th India Council Int. Conf. (INDICON)*, Dec. 2019, pp. 2019–2022.

[37] *Human Protein Atlas Image Classification*. Accessed: Nov. 15, 2021. [Online]. Available: https://www.kaggle.com/c/human-protein-atlas-image-classification
[38] G. Liang, H. Hong, W. Xie, and L. Zheng, "Combining convolutional neural network with recursive neural network for blood cell image classification," *IEEE Access*, vol. 6, pp. 36188–36197, 2018.
[39] Z. Gao, L. Wang, L. Zhou, and J. Zhang, "HEp-2 cell image classification with deep convolutional neural networks," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 2, pp. 416–428, Mar. 2017.
[40] S. Aggarwal, S. Gupta, R. Kannan, R. Ahuja, D. Gupta, S. Juneja, and S. B. Belhaouari, "A convolutional neural network-based framework for classification of protein localization using confocal microscopy images," *IEEE Access*, vol. 10, pp. 83591–83611, 2022.

SONAM AGGARWAL received the bachelor's and master's degrees in electronics and communication engineering from Kurukshetra University, Haryana. She is currently associated with Chitkara University, Punjab. Her research interests include artificial intelligence, biomedical signal processing, machine learning, deep learning, and computer vision.

SAPNA JUNEJA received the master's and Ph.D. degrees in computer science and engineering from Maharshi Dayanand University, Rohtak, in 2010 and 2018, respectively. She is currently working as a Professor with the Department of Computer Science, KIET Group of Institutions, Ghaziabad, India. She has more than 17 years of teaching experience. She has guided several research theses for UG and PG students in computer science and engineering. She has published six patents and various research papers in the renowned national and international journals. Her topic of research is software reliability of embedded system. Her research interests include software engineering, computer networks, operating systems, database management systems, and artificial intelligence. She is a reviewer of several international journals of repute.

JUNAID RASHID received the B.S. and M.S. degrees in computer science from the COMSATS Institute of Information Technology, Wah Campus, Pakistan, in 2014 and 2016, respectively, and the Ph.D. degree in computer science from the University of Engineering and Technology, Taxila, Pakistan, in 2020. Since November 2020, he has been working as an Honorary Senior Postdoctoral Research Fellow at the Center for AI and Data Science, Edinburgh Napier University, U.K., and was awarded the Fellowship, in 2022. He has also worked as a Postdoctoral Researcher at Kongju National University, South Korea, where he is currently a Research Professor. He has published research papers in prestigious journals and conferences. His research interests include data science, machine learning, natural language processing, topic modeling, text mining, information retrieval, big data, pattern recognition, fuzzy systems, medical informatics, biomedical text analytics, and software engineering. He received a fully-funded scholarship for his Ph.D. degree. He has been working as an Academic Editor in *PLOS ONE*. He has served/been serving as a reviewer for various reputed journals and conferences.

DEEPALI GUPTA is currently working as a Research Professor with Chitkara University Research and Innovation Network (CURIN), Chitkara University, Punjab, India. She specializes in software engineering, cloud computing, and genetic algorithms. She has worked with undergraduate students, postgraduate students, and research scholars throughout her career and plans to continue to involve students in her research. She is eager to participate in projects and guide independent student's research. She has published more than 50 research papers in national and international journals and conferences. Based on these areas she has guided many Ph.D. and M.E. scholars. She has worked at various administrative positions like the Principal, the Head (CSE), the Dean Academics, an IBM (Spoc), a Remote Centre Coordinator (IITB), a Coordinator for IITB Spoken Tutorial, an Executive Committee Member at the Computer Science Division of Haryana State Centre (IEI), the President Sangam Kala Group (Kurukshetra, Mohali, and Chandigarh Chapter), a Member of Anti-Ragging Committee, Academic Council, Faculty of Engineering and Technology, and a Board of Management, the Chairperson of SC/ST Cell, Maharishi Markandeshwar University, Sadopur, and the Principal (MMGI, Sadopur). Her research interests include software engineering, cloud computing, machine learning, and blockchain. She is an Active Member of various professional bodies like IEI (India), IETE, and ISTE, apart from being the Editor-in-Chief of *MMU Journal of Management Practices*. She is an editorial board member and a reviewer of various journals.

SHEIFALI GUPTA is currently a Professor with Chitkara University Research and Innovation Network (CURIN), Chitkara University, Punjab Campus, India. She specializes in the area of digital image processing, pattern recognition, machine intelligence, biomedical image processing, agriculture-based image processing, and deep learning. She has published more than 100 research papers and articles in reputed national and international journals and conferences. She has filed 25 patents in the field of image processing and mechatronics. She has conducted different workshops based on modeling real-time video processing system in Simulink using the video and image processing blockset. She has received a Prestigious IRDP Award 2018 for remarkable achievements in teaching, research, and publications.

JUNGEUN KIM received the Ph.D. degree in knowledge service engineering from KAIST.

He is currently an Assistant Professor with Kongju National University (KNU). Before joining KNU, he was a Senior Researcher at the Artificial Intelligence Research Laboratory, Electronics and Telecommunications Research Institute (ETRI). His research interests include data mining, artificial intelligence, big data analysis with distributed processing platforms, and open data platforms.