applied sciences

MDPI

Article

Classification of the Human Protein Atlas Single Cell Using Deep Learning

Tahani Alsubait [1,*,†], Taghreed Sindi [2,†] and Hosam Alhakami [2,†]

[1] Department of Information Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia
[2] Department of Computer Science, College of Computers and Information Systems, Umm Al-Qura University, Makkah 24382, Saudi Arabia
* Correspondence: tmsubait@uqu.edu.sa
† These authors contributed equally to this work.

Abstract: Deep learning has made great progress in many fields. One of the most important fields is the medical field, where we can classify images, detect objects and so on. More specifically, deep learning algorithms entered the field of single-cell classification and revolutionized this field, by classifying the components of the cell and identifying the location of the proteins in it. Due to the presence of large numbers of cells in the human body of different types and sizes, it was difficult to carry out analysis of cells and detection of components using traditional methods, which indicated a research gap that was filled with the introduction of deep learning in this field. We used the Human Atlas dataset which contains 87,224 images of single cells. We applied three novel deep learning algorithms, which are CSPNet, BoTNet, and ResNet. The results of the algorithms were promising in terms of accuracy: 95%, 93%, and 91%, respectively.

Keywords: deep learning; health and safety; human protein atlas; single cell classification

Citation: Alsubait, T.; Sindi, T.; Alhakami, H. Classification of the Human Protein Atlas Single Cell Using Deep Learning. *Appl. Sci.* **2022**, *12*, 11587. https://doi.org/10.3390/app122211587

Academic Editor: Keun Ho Ryu

Received: 14 September 2022
Accepted: 21 October 2022
Published: 15 November 2022

1. Introduction

Deep learning methods [1] are among the most popular methods of classification and prediction [2]. Currently, deep learning algorithms expand in many areas within different sectors such as gaming [3], image classification [4], autonomous cars [5], and image recognition [6], Plant Diseases [7] IoT Scenarios [8], COVID-19 Detection and management [9,10], Architectural Heritage Images [11], among others.

One of the most common areas that use deep learning algorithms is medical image processing because of its complexity, requiring a lot of time to analyze, with the need to process huge amounts of images [12,13]. In this study, we will apply deep learning algorithms to the medical field and in particular to human cells classification [14], by using the public images dataset called the Human Cell Atlas (HCA). The main aim of this study is to classify the structure of the cell into nineteen classes. This classification helps in the distinction of cell types and the changes happening in it [15].

The beginning of dealing with medical images for prediction and analysis was through dealing with the images resulting from computed tomography, X-rays, radiographs, and so on, to identify a patient's problem. However, this was not sufficient for early detection of diseases [16]. Then, the quality of the images that are used in the training of deep learning algorithms was developed and that allowed for dealing with different samples of the human body tissues that contain thousands of cells [17]. This helped to improve the outcomes and results, but it is still difficult to determine the changes that occurred within a single cell. Therefore, the type of images that are used in training algorithms has evolved into single-cell images in order to reveal the cell's structure, the locations of the proteins within it, and any other changes within it; studies have proven the relationship of these changes in the single-cell by discovering diseases early and helping to find the appropriate

treatment [18]. The scope of this research is to integrate deep learning, which is part of artificial intelligence, with the single cell, which is part of biology, with medical images that come to us from the medical field as shown in Figure 1.

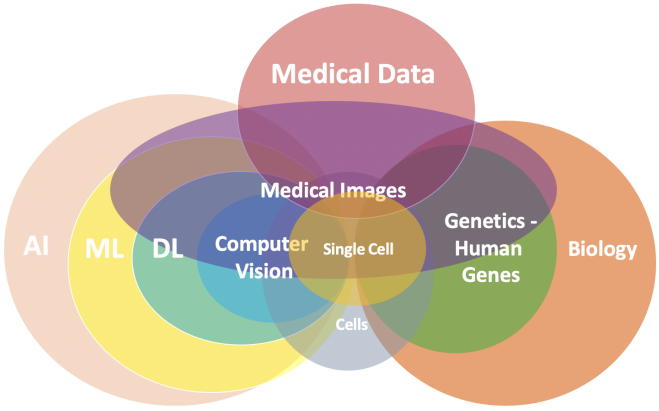

Figure 1. Scope of research.

The contribution of this study is to use a new algorithm called Bottleneck Transformers for Visual Recognition to classify the single-cell for humans. Often, deep learning and computer vision algorithms use batch normalization to obtain the best result [19]. We found a significant improvement on the image classification [20], instance segmentation [21,22], and object detection [23–25] by deep convolutional backbone structures [4,26,27]. However, the architecture of Bottleneck Transformers is derived from backbone architectures. Typically in their architectures, they use multiple layers of 3×3 convolutions [26]. This is because the warp layers effectively aid in computer vision tasks such as object detection, image classification, position segmentation, and key point detection. This architecture helps us to obtain a powerful solution that is able to scale without adding many layers. In addition, in this research, we used another algorithm called the Cross-State Partial Network. This algorithm has special architectures to give good accuracy with less mathematical complexity. This allows anyone who has a computer with a low requirement to use a deep learning algorithm and obtain a good result. Finally, we used the Residual Networks ResNet algorithm on the single-cell dataset with nineteen classes to see how we can enhance the accuracy of calcification. ResNet was ranked as a top 10 accurate algorithm in the list of best algorithms and won in the ImageNet challenge in 2015.

The remainder of this paper organized as follows: Section 2 presents an overview of Cell Classification and Single Cell Classification using deep learning algorithms in the literature review. Section 3 describes the data set used to train the model. Section 4 presents the pre-processing setup of the project. Section 5 provides the methodology conducted in this study. Section 6 presents the results of the algorithm. Finally, Section 7 summarizes this research and proposes future work to enhance the model.

2. Literature Review

There are billions of humans on this earth, and each of them contains trillions of cells that generate huge amounts of medical images [28]. The researchers used artificial intelligence (AI) algorithms to predict information from medical images. Then, they used deep learning models for classification cell images to obtain more data from them. In addition, to obtain higher accuracy, deep learning algorithms were used on a single cell to classify the components of the cell and the location of proteins in it. The following sections reviews some of the articles in this field and Tables 1 and 2 summarizes them.

2.1. Cell Classification

In [29], Sullivan et al. combined two image classification methods for large-scale classification. The dataset used in this paper is public data called Human Protein Atlas (HPA). The first method is a video game called Project Discovery, which helps in the task of image classification for which 322,006 players participated. The second method is used by a deep learning model called the Localization Cellular Annotation Tool (Loc-CAT) [29]. There are 29 classes of proteins in images. After combining these two methods, the accuracy of protein distribution is F1 score of 0.72.

In 2019 [30], a competition was created to develop models for the classification of protein microscope images with a multi-label [31]. Previously, there was a model for classifying the proteins images dependent on cell types of different morphology was called Loc-CAT [29]. However, the performance of this model is less than (0.47) when using the F1 measure. The competition has 2172 teams over a period of three months with a total of 55,213 submissions. The dataset used was 42,774 nonpublic images prepared to be used in this competition. In addition, the teams can use external data such as Human Protein Atlas image collection (HPAv18) contain 78,000 images. There are two challenges in this competition, namely the class imbalance and multi-label problems. The first challenge is the frequencies of classes; the imbalanced such as the most common class, more than other classes, is 'nucleoplasm', and there have been (12,885 images in the training and 31,590 images in HPAv18). The second challenge is the multi-label problem; this means the need to assign the multi-classes to each image. After the final ranking, the F1 scores of the top three teams are 0.593, 0.571, and 0.570 sequentially.

Table 1. Cells classification.

Ref.	Dataset	Algorithm	Classes	Accuracy
[29]	Human Protein Atlas (HPA)	Project Discovery	29	72%
[29]	Human Protein Atlas (HPA)	Localization Cellular Annotation Tool Loc-CAT	29	72%
[30]	42,774 nonpublic images + HPAv18	DenseNet20 + Network size medium	–	59.3%
[30]	42,774 nonpublic images + HPAv18	DenseNet20 + Focused on data preprocessing	–	57.1%
[30]	42,774 nonpublic images + HPAv18	DenseNet20 + Focused on Data augmentation	–	57.0%

2.2. Single Cell Classification

In [32], Durr et al. analyze the high-content screening (HCS) dataset to detect cellular morphology and protein localization. This study suggested three machine learning algorithms that are SVM, LDA, and random forest to classify the features. The result of this model is SVM is 87.6%, the random forest is 89.2%, and LDA is 91.1%. Then, train on Convolutional Neural Net CNN. The result after 100 epochs showed the validation accuracy being on average 0.9313 (SD, 0.0079).

In [33], Meng et al. developed a framework to classify single cells using deep learning. The algorithm used to do the classification was a CNN-based model. In addition, a single-cell image dataset obtained from the Quantitative Phase Imaging QPI system was used and, in the preprocessed phase, the author's additional channel and augmentation and intensity normalization. Then, each pixel in each channel was normalized to be between 0 and 1. After that, each cell imaging with four different fiber coupling angles was represented. Finally, this model can classify different types of single-cell images automated without any human intervention with an accuracy of 96.5%. In addition, this work was supported in part by the NSFC/RGC under Project N_HKU714/13, and GRF 17245716, and the Croucher Innovation Award.

Table 2. Single cell classification.

Ref.	Dataset	Algorithm	Classes	Accuracy
[33]	QPI	CNN	–	96.5%
[32]	BBBC022v14	SVM	4	87.6%
[32]	BBBC022v14	random forest	4	89.2%
[32]	BBBC022v14	LDA	4	91.1%
[32]	BBBC022v14	CNN	4	93.4%
[32]	BBBC022v14	SVM	3	88.6%
[32]	BBBC022v14	random forest	3	88.6%
[32]	BBBC022v14	LDA	3	94.6%
[32]	BBBC022v14	CNN	3	97.3%

3. Dataset

In this study, we will use the public dataset HPA as shown in Figure 2 [34]. There are 87,224 images in train_images with all the images divided into four folders; each folder has a different filter as follows: the protein of interest (green), nucleus (blue), microtubules (red) and endoplasmic reticulum (yellow). In addition, there are 19 labels to predict protein organelle localization as follows: 0. Nucleoplasm, 1. Nuclear membrane, 2. Nucleoli, 3. Nucleoli fibrillar center, 4. Nuclear speckles, 5. Nuclear bodies, 6. Endoplasmic reticulum, 7. Golgi apparatus, 8. Intermediate filaments, 9. Actin filaments, 10. Microtubules, 11. Mitotic spindle, 12. Centrosome, 13. Plasma membrane, 14. Mitochondria, 15. Aggresome, 16. Cytosol, 17. Vesicles and punctate cytosolic patterns, and 18. Negative. In addition, we have a file called train.csv content, the image id, and multi-label classification column. Finally, we have a file called sample_submission.csv; this file is not a multi-label but segments every single cell contained in the image to the class of that cell [35].

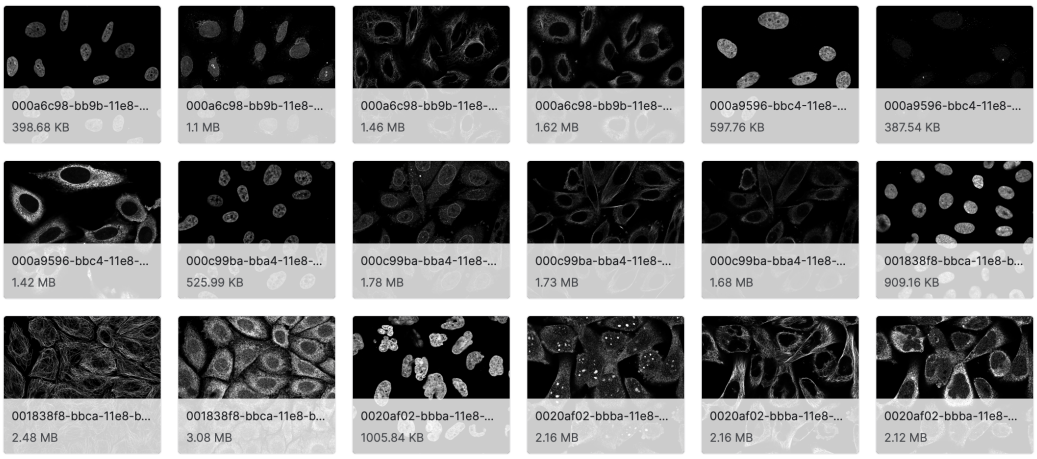

Figure 2. Sample of the dataset [34].

4. Pre-Processing

In this section, we present the preprocessing steps carried out on the data so as to improve the accuracy of algorithms. There are many ways to apply preprocessing steps. In this study, we apply different preprocessing approaches to each algorithm as detailed below.

We use different preprocessing for each algorithm such as cropping at a random size, flipping horizontally with a 0.5 axis, flipping vertically with a 0.5 axis, transposing the image which is similar to rotating the image 90 degrees and cutting out parts of the image to make sure it is still able to identify the key points and changing values of the

hue to incorporate the changes in lighting . . . , etc. All this was completed by utilizing a library in Python called Albumentations for image augmentation. The purpose of image augmentation is to create new training samples from the existing data. This library supports all common computer vision tasks such as classification, semantic segmentation, instance segmentation, object detection, and pose estimation. The library provides a simple unified API to work with all data types such as RBG-images, grayscale images, multispectral images, segmentation masks, bounding boxes, and key points. The library contains more than 70 different augmentations to generate new training samples from the existing data. This data augmentation makes the 2×, 4×, or even 8× times the original data and thus helps our model generalize well and give more suitable predictions.

5. Methodology

Since the ultimate objective was to classify single cells in the Human Protein Atlas (which would be referred to as HPA later) dataset, we had to draw out a comparison between the three different architectures that we would make use of. We were given Red, Green, Blue, and Yellow Image channels for a single record so it was quite evident that we would be using Convolutional Neural Networks as the main architecture for our research. CNNs come with a lot of different variants and architectures, each with its own pros and cons and, in this report, we will discuss how each architecture performed against the validation data. In the following, we present the three architectures that we have worked and trained our model on.

5.1. CSPNet Algorithm

We choose the CSPNet model to apply in this study because this algorithm reduces using the memory and the computation cost and increases the accuracy and speed; this the main aim of CSPNet [36]. The algorithm does that by dividing the feature map into two parts and then merging them through a proposed cross-stage hierarchy. In our CSPNet model, we have used transfer learning and made use of a pre-trained model on the CSPNet to save us time and obtain better results at the same time. In addition, we used the rectified linear activation function The ReLU activation function for the hidden layers of our code for its obvious advantages such as the output will be direct if the input is positive, and, if the input is negative, the output will be zero [37]. In addition, we use this activation function to help our model to learn faster and obtain the best result. Because we have a multiclass classification problem, we used the softmax activation function at the output layer to predict the probability of each label [38]. Then, we used a Python library to find the best learning rate for our algorithm.

5.2. ResNet Algorithm

We chose this algorithm because it was ranked top 10 accuracy algorithms in the list of best algorithms. In this project, we made use of PyTorch and its internal dependencies for this project including the Lightning module [39]. There are more features for use ResNet such as it allows us to solve the problem of vanishing gradients; this was a very big problem for deep deep neural networks. We made use of pre-trained ResNet weights online that were trained by different researchers from across the world and used transfer learning to get it accomplished. In a simple way, we downloaded the weights of ResNet50 via the PyTorch library. Our model has 50 layers. We froze the pre-trained weights and added a simple convolution and pooling layer.

5.3. BoTNet Algorithm

The BoTNet is a conceptually simple yet powerful backbone architecture that incorporates self-attention for multiple computer vision tasks by just replacing the spatial convolutions with global self-attention [40] in the final three bottleneck blocks of a ResNet and no other changes. The architecture of our model contains three layers. If you notice, you will find a 512 neuron-sized Multi-Head Self-Attention MHSA [41] layer that stands for

Multi Headed Self Attention. This was the breakthrough in achieving even higher accuracy. This MHSA layer was actually a 3×3 Convolution layer in ResNet, but in BoTNet, it was replaced by an MHSA layer to improve performance. Since we have used BoTNet, we made use of transfer learning from the pre-trained BoTNet to have the weighted parameters. Then, we added our own layer on top of it in a sequential manner. We added a convolution layer, an average pooling layer, and then flattened it out and passed it to a linear layer that predicted our outputs. For the hyperparameters, we have used the learner PyTorch module, which helped us find the best learning rate for our program which turned out to be 0.030.

6. Results and Evaluation

In this section, we will present the results of the three algorithms that we applied in this study. Then, we will compare them to see which algorithm gives the best accuracy for classifying the single cell.

To start with, we tried the CSPNet algorithm on the original dataset before augmentation steps discussed in previous sections, and we obtained a good result compared to existing research, which was 84% for the accuracy measure, but due to the sensitivity of the classification task at hand, we aimed to improve the results by applying additional preprocessing steps as explained earlier.

The result of the CSPNet algorithm after prepocessing and hyperparameters' setting is presented below. We ran it against nine Epochs from 0–8 inclusive. The accuracy in Epoch 0 was 94.2%, and the accuracy improved in Epoch 8 and reached 95%; to represent the relation between training and validation to clarify, there is no overfitting as shown in Figure 3.

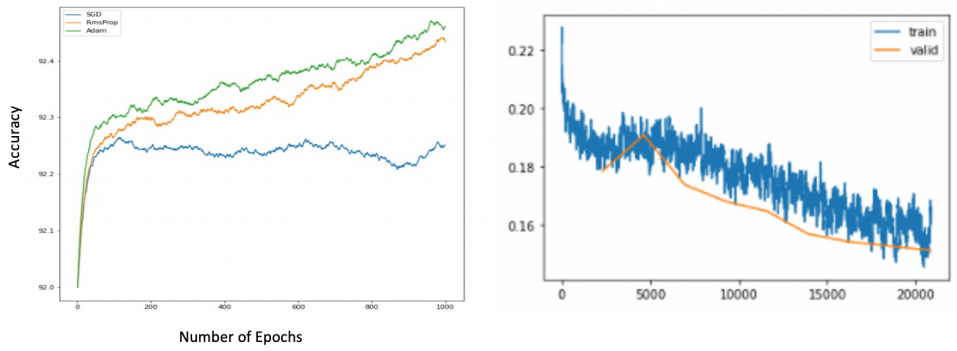

Figure 3. The accuracy of CSPNet.

When we ran the ResNet model for 340 Epochs, our model reached a validation accuracy of 0.91, which is excellent as shown in Figure 4. It was trained on Kaggle TPU. If we had not used transfer learning from ResNet, the accuracy would not have been that much better. Utilizing the pre-training model on images helped us reach a 90%+ accuracy on the validation set. We modeled to a total of 25.6 Million params, and it was trained on Kaggle 8 cored Kaggle TPU.

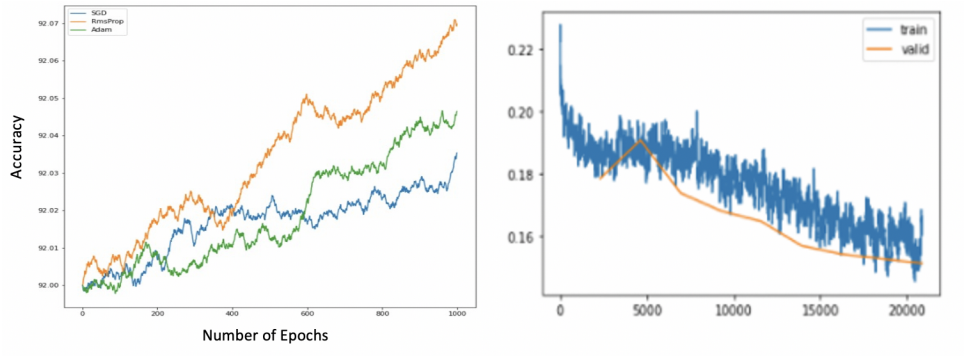

Figure 4. The accuracy of ResNet.

When we ran the BoTNet model, the accuracy we have is 93.8% in second epochs. In addition, there is no overfitting as shown in Figure 5. The accuracy we reached is quite promising for this model with only a thousand records. The idea of self-attention definitely helped us achieve better results. For every channel, using the idea of self-attention, the accuracy, therefore, improved from 0.91 to 0.937, although, if we had even higher computational power, we could have possibly increased the accuracy by training on even more epochs.

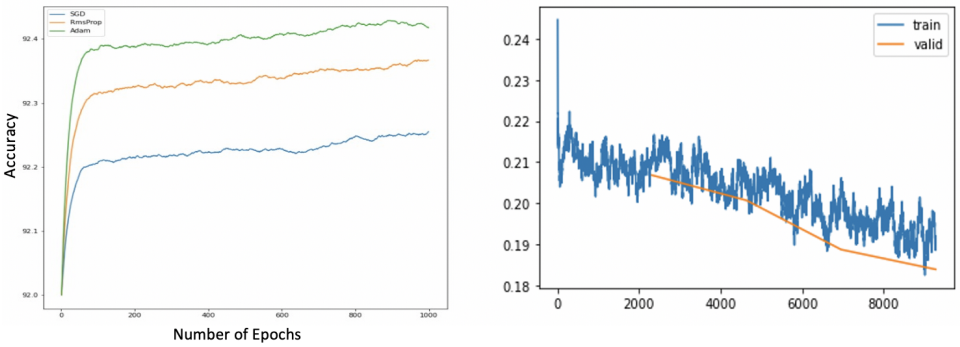

Figure 5. The accuracy of BoTNet.

After that, we run the three algorithms on the same training dataset with the different preprocessing steps. We ranked the models on the basis of model loss as follows: CSPNet, BoTNet, and ResNet as shown in Figure 6.

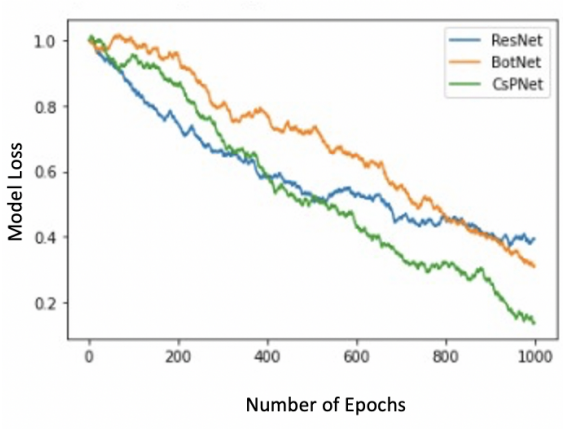

Figure 6. Model loss of CSPNet, BoTNet, and ResNet.

CSPNet outperformed every other model with a accuracy of 95%. This model uses an optimization network that allows the computations to be reduced by 20%, thus giving aid to

higher accuracy. Since CSPNet was implemented on top of ImageNet, it has already made use of transfer learning, and it has stable pre-trained parameters from the image net model to give it an accuracy boost. We have also used image augmentation to increase the data and improve predictions. The second on the list was BoTNet, which was developed by a group of researchers specifically for these kinds of situations and recurrent neural networks. This allows us to improve in situations where we have multiple channels or words, and they impact each other. In our case, we had different channels and we were dealing with them separately but BoTNet allowed us to put an impact of one channel on the other one and combined their effect using a phenomenon called self-attention of the transformer neural networks. That is how it was able to perform so well and give an accuracy of 93.9%. Finally, ResNet came in third on our list with accuracy being 0.91%. ResNet is in itself a good model and has won the 2015 image net competition, and the technology it introduced called skip connections has greatly added to the machine computer vision and deep learning community. In addition, it ranks in the top 10 image classification algorithms on the Imagenet accuracy list. The concept of BoTNet self-attention is somehow related to the concept of ResNet skip connections. Nevertheless, it outperformed other models and was able to outperform it for the obvious reasons that they were making use of the pre-trained weights of ResNet and adding further convolutional and pooling layers on top of it. Thus, no doubt they would be better but it does not diminish the importance of ResNet which has been around for years and is always performing well. We summarize the results of three algorithms in this study in the following Table 3. Finally, although one network performed better than the others in this case, all of these models have their own usage and specialties, and there is no golden model to pick for every problem that would give the best accuracy in all the datasets and problems.

Table 3. The result of three algorithms.

Algorithm	Number of Epochs	Train Loss	Valid Loss	Accuracy
CSPNet	9	0.16	0.1528	95%
BoTNet	4	0.19	0.2007	93%
ResNet	340	0.22	0.1528	91%

7. Conclusions and Future Work

In this research, we used deep learning algorithms because it excels at dealing with large and complex data. In addition, we chose three algorithms of deep learning to classify single cells into 19 classes. As it is possible that one image contains more than one class, this will be multi labeling. In addition, we used single-cell classification because that gives us better results than the classification of human tissue, which contains thousands of cells. The algorithms we used were CSPNet, BoTNet, ResNet and we applied them to the Human Protein Atlas dataset with a size of 158.36 GB. After that, we apply different preprocessing approaches to data for each algorithm. The accuracy we obtained from classification of the single-cell image was 95% for the CSPNet algorithm, while the algorithm that obtained the least accuracy was ResNet.

In future work, we seek to increase the accuracy of single-cell classification by trying different algorithms and preprocessing. We will also study the impact of quality or resolution of images on the results. For example, in the used dataset, there were images with different resolutions such as 1728 $\times$ 1728, 2048 $\times$ 2048, and 3072 $\times$ 3072. Further experiments are needed to show the impact of good/bad resolutions or the type of noise in the images on the overall results. We also aim to increase the number of classes in single-cell to reach highly accurate results, which would help discover diseases early and also to help innovate drugs to treat these diseases, in addition to determining the expected disease based on the changes taking place inside the cell. We would like to specifically work on early detection of the two types of cancer that are most common in the world, namely breast

and lung cancer. Cancer is a major cause of death worldwide and caused about 10 million deaths in 2020, where the number of people with breast cancer was 2.26 million cases and lung cancer 2.21 million cases, according to the statistics of the World Health Organization. We will do this by collaborating with specialists in the medical field, specifically cancer, and cellular pathologists.

Author Contributions: All authors have contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (20UQU0066DSR).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
2. Buettner, R.; Bilo, M.; Bay, N.; Zubac, T. A Systematic Literature Review of Medical Image Analysis Using Deep Learning. In Proceedings of the 2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA), TBD, Malaysia, 17–18 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–4.
3. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef] [PubMed]
4. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
5. Bojarski, M.; Del Testa, D.; Dworakowski, D.; Firner, B.; Flepp, B.; Goyal, P.; Jackel, L.D.; Monfort, M.; Muller, U.; Zhang, J.; et al. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
6. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
7. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. *Plants* **2019**, *8*, 468. [CrossRef] [PubMed]
8. Heidari, A.; Jabraeil Jamali, M.A.; Jafari Navimipour, N.; Akbarpour, S. Deep Q-Learning Technique for Offloading Offline/Online Computation in Blockchain-Enabled Green IoT-Edge Scenarios. *Appl. Sci.* **2022**, *12*, 8232. [CrossRef]
9. Rehman, A.; Iqbal, M.A.; Xing, H.; Ahmed, I. COVID-19 Detection Empowered with Machine Learning and Deep Learning Techniques: A Systematic Review. *Appl. Sci.* **2021**, *11*, 3414. [CrossRef]
10. Heidari, A.; Jafari Navimipour, N.; Unal, M.; Toumaj, S. Machine learning applications for COVID-19 outbreak management. *Neural Comput. Appl.* **2022**, *34*, 15313–15348. [CrossRef]
11. Llamas, J.; M. Lerones, P.; Medina, R.; Zalama, E.; Gómez-García-Bermejo, J. Classification of Architectural Heritage Images Using Deep Learning Techniques. *Appl. Sci.* **2017**, *7*, 992. [CrossRef]
12. Fourcade, A.; Khonsari, R. Deep learning in medical image analysis: A third eye for doctors. *J. Stomatol. Oral Maxillofac. Surg.* **2019**, *120*, 279–288. [CrossRef] [PubMed]
13. Kumar, E.S.; Bindu, C.S. Medical image analysis using deep learning: A systematic literature review. In Proceedings of the International Conference on Emerging Technologies in Computer Engineerin, Jaipur, India, 1–2 February 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 81–97.
14. Schlüter, J.P.; Czuppon, P.; Schauer, O.; Pfaffelhuber, P.; McIntosh, M.; Becker, A. Classification of phenotypic subpopulations in isogenic bacterial cultures by triple promoter probing at single cell level. *J. Biotechnol.* **2015**, *198*, 3–14. [CrossRef] [PubMed]
15. Glaros, A. *Data-Driven Definition of Cell Types Based on Single-Cell Gene Expression Data*; Uppsala University: Stockholm, Sweden , 2016.
16. Rajpurkar, P.; Irvin, J.; Zhu, K.; Yang, B.; Mehta, H.; Duan, T.; Ding, D.; Bagul, A.; Langlotz, C.; Shpanskaya, K.; et al. Chexnet: Radiologist-level pneumonia detection on chest X-rays with deep learning. *arXiv* **2017**, arXiv:1711.05225.
17. Elazab, N.; Soliman, H.; El-Sappagh, S.; Islam, S.; Elmogy, M. Objective Diagnosis for Histopathological Images Based on Machine Learning Techniques: Classical Approaches and New Trends. *Mathematics* **2020**, *8*, 1863. [CrossRef]
18. Poulin, J.F.; Gaertner, Z.; Moreno-Ramos, O.A.; Awatramani, R. Classification of midbrain dopamine neurons using single-cell gene expression profiling approaches. *Trends Neurosci.* **2020**, *43*, 155–169. [CrossRef]
19. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 7–9 July 2015; pp. 448–456.
20. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]

21. Hariharan, B.; Arbeláez, P.; Girshick, R.; Malik, J. Simultaneous detection and segmentation. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 297–312.
22. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
23. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
24. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
25. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
26. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
27. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
28. Ouyang, W.; Zimmer, C. The imaging tsunami: Computational opportunities and challenges. *Curr. Opin. Syst. Biol.* **2017**, *4*, 105–113. [CrossRef]
29. Sullivan, D.P.; Winsnes, C.F.; Åkesson, L.; Hjelmare, M.; Wiking, M.; Schutten, R.; Campbell, L.; Leifsson, H.; Rhodes, S.; Nordgren, A.; et al. Deep learning is combined with massive-scale citizen science to improve large-scale image classification. *Nat. Biotechnol.* **2018**, *36*, 820–828. [CrossRef]
30. Ouyang, W.; Winsnes, C.F.; Hjelmare, M.; Cesnik, A.J.; Åkesson, L.; Xu, H.; Sullivan, D.P.; Dai, S.; Lan, J.; Jinmo, P.; et al. Analysis of the human protein atlas image classification competition. *Nat. Methods* **2019**, *16*, 1254–1261. [CrossRef]
31. Tsoumakas, G.; Katakis, I. Multi-label classification: An overview. *Int. J. Data Warehous. Min. (IJDWM)* **2007**, *3*, 1–13. [CrossRef]
32. Dürr, O.; Sick, B. Single-cell phenotype classification using deep convolutional neural networks. *J. Biomol. Screen.* **2016**, *21*, 998–1003. [CrossRef] [PubMed]
33. Meng, N.; Thus, H.K.; Lam, E.Y. Computational single-cell classification using deep learning on bright-field and phase images. In Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 8–12 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 190–193.
34. kaggle Human Protein Atlas—Single Cell Classification: Challenge. Dataset Sample—Microscope Images. 2020. Available online: https://www.kaggle.com/philculliton/hpa-2020-16bit-training-set (accessed on 11 May 2022).
35. kaggle Human Protein Atlas—Single Cell Classification: Challenge. Data Description. 2021. Available online: https://www.kaggle.com/c/hpa-single-cell-image-classification/data?select=train (accessed on 11 May 2022).
36. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
37. Eckle, K.; Schmidt-Hieber, J. A comparison of deep networks with ReLU activation function and linear spline-type methods. *Neural Netw.* **2019**, *110*, 232–242. [CrossRef] [PubMed]
38. Dunne, R.A.; Campbell, N.A. On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function. In Proceedings of the 8th Australian Conference on the Neural Networks, Melbourne, Australia, 11–13 June 1997; Citeseer: State College, PA, USA, 1997; Volume 181, p. 185.
39. McMahon, D.J.; Brown, R.J. Evaluation of Formagraph for comparing rennet solutions. *J. Dairy Sci.* **1982**, *65*, 1639–1642. [CrossRef]
40. Shen, Z.; Bello, I.; Vemulapalli, R.; Jia, X.; Chen, C.H. Global Self-Attention Networks for Image Recognition. *arXiv* **2020**, arXiv:2010.03019.
41. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv* **2019**, arXiv:1905.09418.