big data and cognitive computing

MDPI

*Article*

# EffResUNet: Encoder Decoder Architecture for Cloud-Type Segmentation

**Sunveg Nalwar [1,*], Kunal Shah [1], Ranjeet Vasant Bidwe [2,*], Bhushan Zope [2], Deepak Mane [3], Veena Jadhav [4] and Kailash Shaw [2]**

[1] SCTR's, Pune Institute of Computer Technology, Pune 411043, India
[2] Symbiosis Institute of Technology, Pune (SIT), Symbiosis International (Deemed) University (SIU), Pune 412115, India
[3] Department of Computer Engineering, JSPM's Rajarshi Shahu College of Engineering, Pune 411033, India
[4] Department of Computer Engineering, Bharati Vidyapeeth (Deemed to Be University), College of Engineering, Pune 411043, India
* Correspondence: sunvegnalwar7@gmail.com (S.N.); ranjeet.bidwe.phd2021@sitpune.edu.in (R.V.B.)

**Abstract:** Clouds play a vital role in Earth's water cycle and the energy balance of the climate system; understanding them and their composition is crucial in comprehending the Earth–atmosphere system. The dataset "Understanding Clouds from Satellite Images" contains cloud pattern images downloaded from NASA Worldview, captured by the satellites divided into four classes, labeled Fish, Flower, Gravel, and Sugar. Semantic segmentation, also known as semantic labeling, is a fundamental yet complex problem in remote sensing image interpretation of assigning pixel-by-pixel semantic class labels to a given picture. In this study, we propose a novel approach for the semantic segmentation of cloud patterns. We began our study with a simple convolutional neural network-based model. We worked our way up to a complex model consisting of a U-shaped encoder-decoder network, residual blocks, and an attention mechanism for efficient and accurate semantic segmentation. Being an architecture of the first of its kind, the model achieved an IoU score of 0.4239 and a Dice coefficient of 0.5557, both of which are improvements over the previous research conducted in this field.

**Keywords:** semantic segmentation; encoder-decoder network; satellite images; clouds

check for updates

**Citation:** Nalwar, S.; Shah, K.; Bidwe, R.V.; Zope, B.; Mane, D.; Jadhav, V.; Shaw, K. EffResUNet: Encoder Decoder Architecture for Cloud-Type Segmentation. *Big Data Cogn. Comput.* **2022**, *6*, 150. https://doi.org/10.3390/bdcc6040150

Academic Editor: Salvador García López

Received: 8 October 2022
Accepted: 30 November 2022
Published: 7 December 2022

## 1. Introduction

Climate change has been at the top of our minds and the forefront of crucial political decision making for many years [1,2]. Identifying and grouping clouds enables scientists to develop more accurate global climate models that could help predict the path of global warming and how our planet might change over time [3]. Machine learning can assist this personnel by reducing the time and effort needed to identify the cloud patterns. The capacity to do so efficiently can lead to more accurate weather forecasting. Semantic segmentation in very high resolution (VHR) aerial photography is becoming more important for applications such as road extraction, urban planning, and land cover categorization. It plays a major role in autonomous driving [4,5]. Semantic scene segmentation has witnessed a tremendous breakthrough over the years [6,7]. Recently, it has become an influential research domain with deep learning [8]. Despite numerous breakthroughs in recent years, scene interpretation in complicated real-world circumstances remains a difficult challenge compared to human performance. Before the development of CNN-based systems, semantic picture segmentation methods relied on hand-crafted features and classical classifiers. Few researchers have also tried to tackle the problem using the swarm intelligence algorithm [9]. However, because CNNs have demonstrated their efficacy in image classification, they are also employed as the backbone for feature extraction in semantic segmentation tasks. Convolutional neural networks lower the input resolution by 32 times to produce a high-level feature map representing the original image. This minimal feature map is handy for image classification when there is just one dominating item in the image. CNNs have outperformed humans in this image classification task [10]. However, the CNN performance in the segmentation job is not much lauded because the spatial information required to analyze the complex features in the image is lost in small feature maps. Previous methodologies involved using FCNs and robust UNet architecture [11] for segmentation, which has improved over time through experiments with its modules. We propose combining the effectiveness of EfficientNet [12] as a pre-trained encoder to extract high-level features along with the residual block decoder in a UNet inspired architecture powered by an attention mechanism [13] for creating fine segmentation maps to overcome this challenge. In this study, we present an approach for semantic segmentation of satellite images of clouds to help in better climate prediction, which could help us in the following ways:

1. Accurate cloud predictions can help in predicting rainfall and climate early, which will help the farmers to take actions accordingly.
2. Climate preparedness is a thing which can help people to prepare for a calamity beforehand, and such predictions are even more accurate if the particular cloud types are known.

Our additional contributions could be listed as follows:

1. We implemented efficient pre-processing and post-processing techniques using Albumentations along with Test Time Augmentation (TTA) to help achieve state-of-the-art results with limited resources.
2. We were able to achieve the highest accuracy so far by establishing a segmentation model architecture that combines the advantages of AttentionUnet [14], EfficientUNet [15], and Residual UNet [16] into a single architecture with a focus on the decoder path to obtain improved results.
3. We suggest using techniques such as the attention mechanism and leveraging transfer learning to achieve state-of-the-art results in less time efficiently.

The rest of the paper is structured as follows: the task-related work along with some of the previous methodologies is discussed in Section 2. Further, Section 3 talks about the dataset used and the data-processing techniques applied during training and testing, while the modules used in our architecture are listed in Section 4. The proposed model architecture is explained in Section 5. The results obtained are presented in Section 6, while the paper is concluded with a discussion on future work in Section 7.

## 2. Related Work

The proposed model's prime module is based on the encoder–decoder architecture. This architecture was chosen because the pictures have specific cloud patterns and context needed to be taken into consideration as there is a high probability of similar pattern of clouds being grouped together. Standard encoder-decoder architectures such as UNet are well-trained, proven to be efficient for such tasks, and are essential for extracting the underlying context-rich information from the images.

### *2.1. Standard Encoder–Decoder Network Architecture (UNet)*

Semantic segmentation can be easily performed using the simplest encoder–decoder network– UNet, a modification of fully connected networks (FCNs) [17]. Such a network consists of a contracting path and an expanding path. These networks use shortcuts or skip connections between the encoder and the decoder. The contracting course is called the encoder, a typical CNN. It is used for extracting the features and progressively down-sampling the original images to obtain their feature maps. This "encoding" can be performed by training the network from scratch or using pre-trained state-of-the-art CNNs, such as ResNet, InceptionResNetV2, MobileNet, or EfficientNet [18,19]. The decoder, an expanding path, concatenates the features coming through the skip connections from the encoder part and sampling the feature map to bring the image back to the original dimensions and output the segmented image. These skip connections provide context to the decoder, so the spatial information from the encoder remains intact.

### *2.2. Previous Methodologies*

Although there were approaches in the past that dealt with image segmentation and classification [20], shallow network architectures were used for the task. It was after the introduction of CNNs that the results were significantly reflected. Evolving from fully connected networks to encoder-decoder architectures, such as UNet, was a major architectural breakthrough in this field, and since then, it continues to be a domain open to further enhancements. The segmentation task proved out to be very helpful in the medical field; the UNet architecture was first implemented as an application in this field. However, these plain networks take up much time for training compared to the CNNs. Hence, vanilla UNets trained on major datasets, such as Imagenet, gained popularity as pre-trained models that are being used as encoders. The pre-trained models save time and resources and leverage transfer learning, improving the performance of other models instead of training them from scratch.

Table 1 highlights some of the major approaches that have been taken by the researchers and were majorly referred by us to arrive at our novel implementation of the idea.

**Table 1.** Previous Methodologies.

| **Title** | **Remark** |
|---|---|
| Fully Convolutional Networks for Semantic Segmentation [17] | The first approach to introduce the family of CNNs in the field of semantic segmentation. However, the loss of spatial features was an issue besides considerable training time. |
| U-Net: Convolutional Networks for Biomedical Image Segmentation [11] | Novel encoder–decoder architecture in image segmentation, usually performs better with pre-trained encoders. Feature extraction is the major work performed by this architecture. |
| Residual U-Net for Retinal Vessel Segmentation [16] | UNet with residual blocks improved results significantly and increased the convergence without tampering with the spatial information. |
| Attention U-Net: Learning Where to Look for the Pancreas [14] | Attention mechanisms have always helped in learning the patterns that vary in shape and size. The mechanism eliminates the need for localizing the objects and allows the model learn by itself as to which parts to focus on. |
| Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment [15] | This architecture explores the advantage of EfficientNet, using the method of compound scaling combined with UNet, works as a better feature extractor. |

## 3. Dataset and Data Processing

### *3.1. Dataset*

The dataset “Understanding Clouds from Satellite Images” contains images downloaded from NASA Worldview. The dataset contains 22,184 images belonging to 4 classes with 5546 images each. The images are classified into these four classes with label names: Fish, Flower, Gravel, and Sugar. One example of each pattern is shown in Figure 1. Three locations were chosen, covering 21° longitude and 14° latitude. The true-color photographs were captured by TERRA and AQUA, two polar-orbiting satellites that pass over the exact location once a day. Because the imager (MODIS) onboard these satellites has a modest footprint, an image from two orbits was stitched together. The area that two subsequent orbits did not cover is indicated in black. The labels were designed as part of a crowd-sourcing project [21] at Hamburg’s Max-Planck-Institute for Meteorology and Paris’s Laboratoire de Météorologie Dynamique. After eliminating any black-band regions from the areas, ground truth was found by adding the areas designated for that picture. Even the human labeling process contained variations in masks because it is a daunting task to recognize cloud patterns given the area that they spread over. The union of all labeled covers was considered for building the dataset. In the case of numerous non-contiguous sections of the same formation in a picture, the segmented masks for each cloud formation label were encoded into a single row. The ground truth masks were provided in an encoded RLE format as a separate file. An image containing ground truth masks is shown in Figure 2.

Sugar pattern: dusting of wonderful clouds, little evidence of self-organization. Flower pattern: large-scale stratiform clouds feature bouquets appearing well-separated from each other. Fish pattern: large-scale skeletal networks of clouds separated from other cloud forms. Gravel pattern: meso-beta lines or arcs defining randomly interacting cells with intermediate granularity.

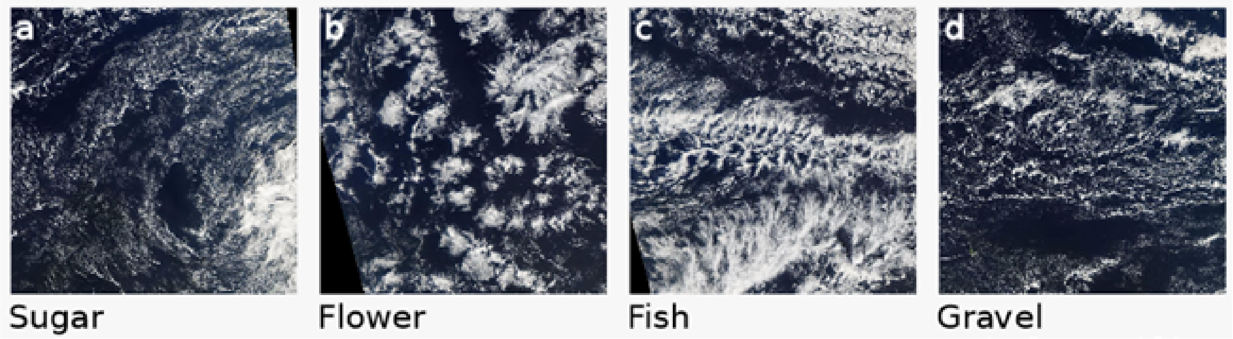


**Figure 1.** Cloud pattern dataset containing patterns of Sugar (**a**), Flower (**b**), Fish (**c**) and Gravel (**d**).

### *3.2. Pre-Processing*

The segmentation of objects containing repeated patterns is a complex task, and in such cases, the more the data, the better is the outcome. In the dataset, we have 5516 images belonging to each class, which is a fairly low amount of images; hence, as a part of preprocessing, we applied augmentation to the dataset. Memory is always a constraint for augmentation using various techniques as these are high-resolution images. This is why we opted for Albumentations [22], a real-time augmentation library that generates high-quality transformed images. For this task, the pipeline we used consisted of horizontal and vertical shifts, each with a probability of applying the transform of 0.5, followed by a geometric transform using ShiftScaleRotate. Training and validation data were split into batches of size 8, and the input images of size 1400 × 2100 pixels were reshaped to 320 × 480 pixels to make them easier to deal with. The ground truth segmentation masks were provided along with the data in run-length-encoding (RLE) format, which is an efficient encoding style for storing the location of covers in compressed form. An extra step in preprocessing involved the conversion of these RLEs to equivalent masks to prepare masked images for training as shown in Figure 2.

#### Post-Processing

Post-processing is a crucial step in wrapping the model. It is equally important to get rid of errors generated during prediction in the previous stage of model building. This step ensures that all the small masks below a minimum size are removed, and a certain threshold is set to decide the segmentation at a pixel level. We even observed that using test time augmentation (TTA) improves the score significantly. TTA is a technique that augments the images at test time and predicts the segmentation map on all those augmented images. The predictions are then aggregated and averaged to obtain the best result [7,23].

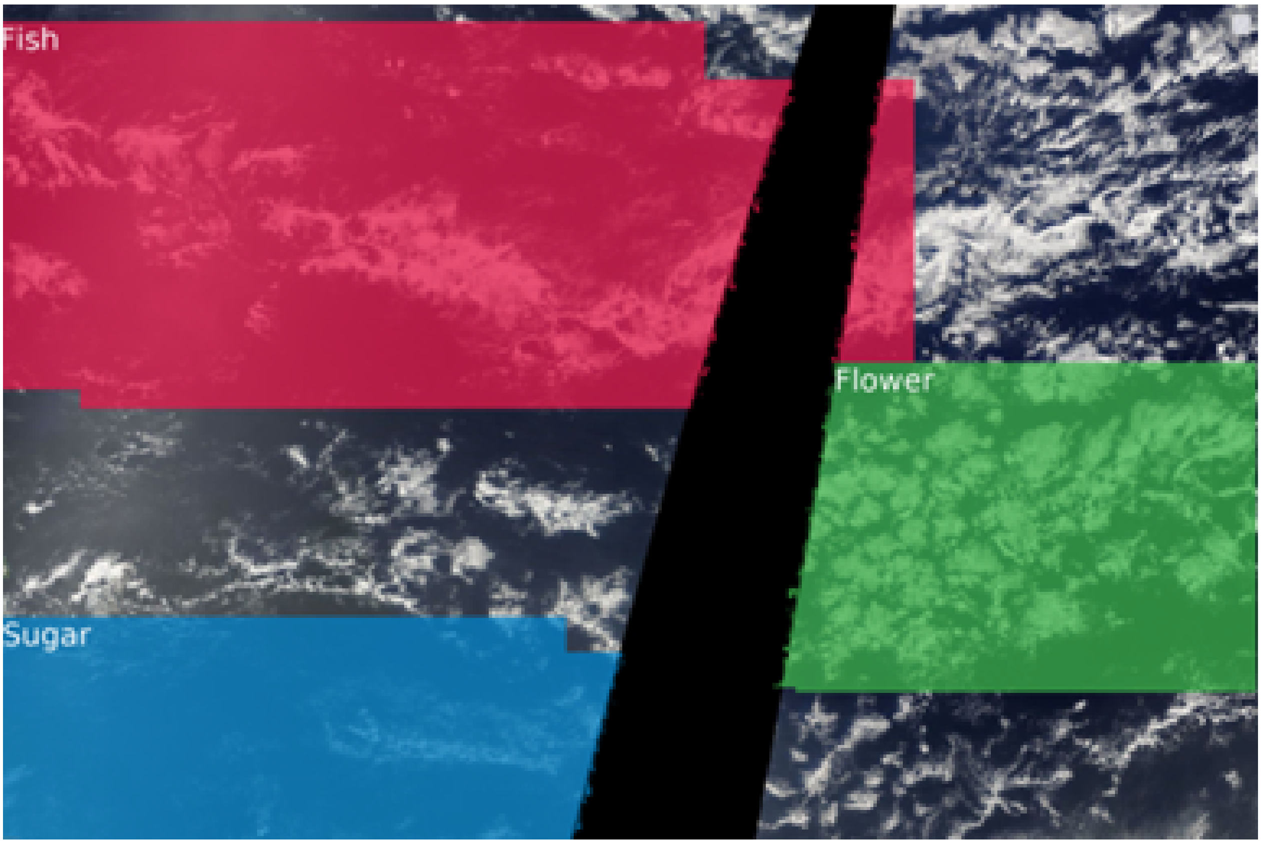


**Figure 2.** Image of masked multi-label cloud patterns.

## 4. Modules Used

The proposed architecture consists of a U-shaped encoder-decoder network. The blocks we used are as follows:

1. EfficientNet encoder;
2. Residual block decoder;
3. Attention mechanism.

### 4.1. *EfficientNet*

Convolutional neural networks perform better when scaled in either width, depth, or resolution. CNNs such as ResNet34 can be scaled to ResNet101 by adding more layers to the network. However, this needs extra resources to afford this kind of scaling. If not optimally maintained, the ratio of scaling in terms of network width, depth, and resolution might not give the best results or could even become computationally expensive. The method to scale up is also not concrete yet. To tackle this issue, a new family of models was invented called EfficientNets. These are 8.4 times smaller and 6.1 times faster than the existing ConvNets [15]. Other state-of-the-art CNNs scale the network using arbitrary constants through trial and error or grid search, whereas EfficientNets use the method of compound scaling [24] that works much more effectively since the scaling coefficients are mathematically fixed. For example, suppose we want to use $2^N$ times more computational resources. In that case, we can increase the network depth by $\alpha^N$, width by $\beta^N$, and image size by $\gamma^N$ , where $\alpha, \beta, \gamma$ are constant coefficients determined by a small grid search on the original miniature model. Based on these coefficients, these networks have eight variants ranging from B0 to B7. Out of these, we found the use of EfficientNetB0 to be the best due to its simplistic nature. We used EfficientNetB0 as the encoder using transfer learning that was pre-trained on the ImageNet dataset. Other variants tend to take up more training time, delivering the same or poorer results. These networks perform very well with transfer learning, making it easier for us to incorporate them into the encoder.

### *4.2. Residual Blocks*

Intense convolutional neural networks require much time for training and heavy resources for computing. This was solved efficiently by using residual blocks that use skip connections, thereby skipping the layers [25]. A residual block contains two connections, one that undergoes multiple convolutions, batch normalization and other linear functions, and another that skips over all these functions. After each block, the outputs of both connections are added. This allows the network to learn quickly and more effectively. Figure 3 shows the residual block used in our model with input x undergoing series of operations denoted by F(x).

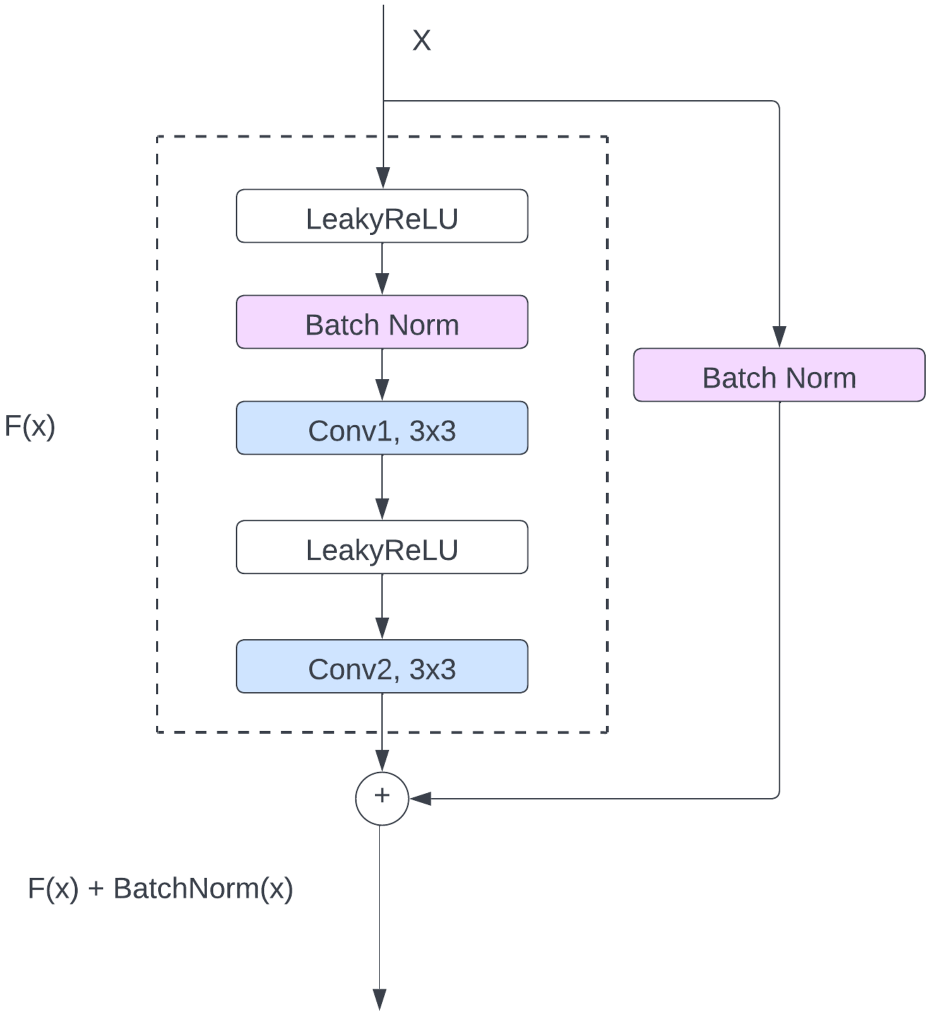


**Figure 3.** Residual block.

### *4.3. Attention Mechanism*

In aerial satellite imagery, you can expect irrelevant information, including pictures of the satellite parts or pictures containing just a plain blue sky. A solution is to apply multiple object localization models followed by segmentation to focus on essential elements. However, this could be a heavy task and can be eliminated by using attention gates implemented in association with residual blocks [26]. The computation is briefly explained further. The attention block has two inputs, the gating signal (g) and input x from the skip connection, see Figure 4. The gating signal (g) has better feature representation, as it is from a deeper part of the network, whereas the input x is rich in spatial information since it is from the earlier layers of the network. The gating signal has half the number of features as that of x. Hence, we need to bring them back to the same shape. To do so, we add convolutional blocks to each. Let us call the convolutional block added to the gating signal

($\phi_g$). This is a $1 \times 1$ Conv2D layer with a stride of (1, 1) kernel size of $1 \times 1$. This will make the number of features in g equal to that in x. Similarly, we add a $1 \times 1$ convolution block $\theta_x$ with a stride of (2, 2) so that all the dimensions in both blocks are equal. Now the outputs from $\phi_g$ and $\theta_x$ are concatenated. This allows the model to extract only the relevant features because the aligned weights become higher than the unaligned weights. After joining, they undergo the ReLU activation function followed by another convolutional block ($\psi$) having the number of filters equal to 1. This is the weight obtained by the mechanism so far. Since this value can range from zero to infinity, we need to bring them back between zero and one. To achieve this, we applied a sigmoid activation function. The obtained result was then up-sampled back to the original size of input x to multiply it with x in an element-wise fashion. This final vector is the scaled version based on feature relevance.

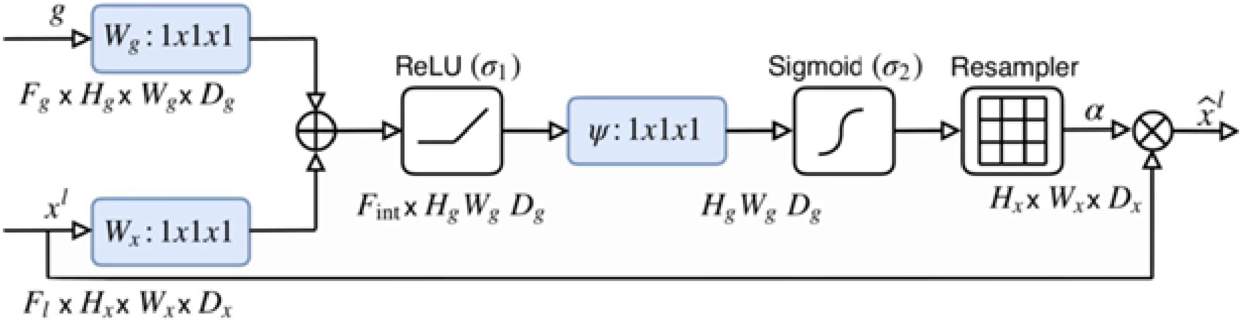


**Figure 4.** Attention mechanism.

## 5. Proposed Architecture

### *5.1. Encoder-Decoder Network*

For the downsampling path (left half), as shown in Figure 5, different pre-trained encoders such as ResNet34, InceptionResNetV2, MobileNet, and EfficientNets were tried using Transfer learning. Out of these, EfficientNetB0 performed pretty well and was selected as the best. EfficientNetB0-B3 took around 6 h to train for 30 epochs whereas the higher variants of EfficientNets (B4 to B7) took more than 8 h to train, yet did not improve the accuracy. This downsampling block consisted of EfficientNet output followed by LeakyReLU, having an alpha of 0.1, and then a $2 \times 2$ max pooling operation. We observed that the shallow layers require a lower dropout rate, as it contains more contextual information, and as we go deeper, the dropout rate should be increased to prevent overfitting. Hence, a dropout rate of 0.1 was fixed for the first block and was increased to 0.25 for the deeper blocks in the network. Different dropout rates were experimented with, and these values were finally chosen as optimal rates. The Dice coefficient was noted for different dropout rates for the first block as shown in Table 2. The dropout rates mentioned in Table 2 are only for the first block and for deeper layers, it was set to 0.25 after similar experimentation. On the other hand, in the up-sampling path (right half), the use of residual blocks along with the attention mechanism is suggested. The output from the previous block of the decoder is up-sampled and kept aside for concatenation with the output from the attention block later. This previous block output is provided as a gating signal and the skip connection from the encoder as inputs to the attention block. In the final upsampling block, the original dimensions are recovered, and a segmented map for four classes with a sigmoid activation function is produced.

**Table 2.** First block dropout and corresponding Dice coefficient score (these scores were before the post-processing techniques).

| Dropout | 0.05 | 0.1 | 0.15 | 0.2 |
|---|---|---|---|---|
| Dice coefficient | 0.509 | 0.518 | 0.503 | 0.496 |

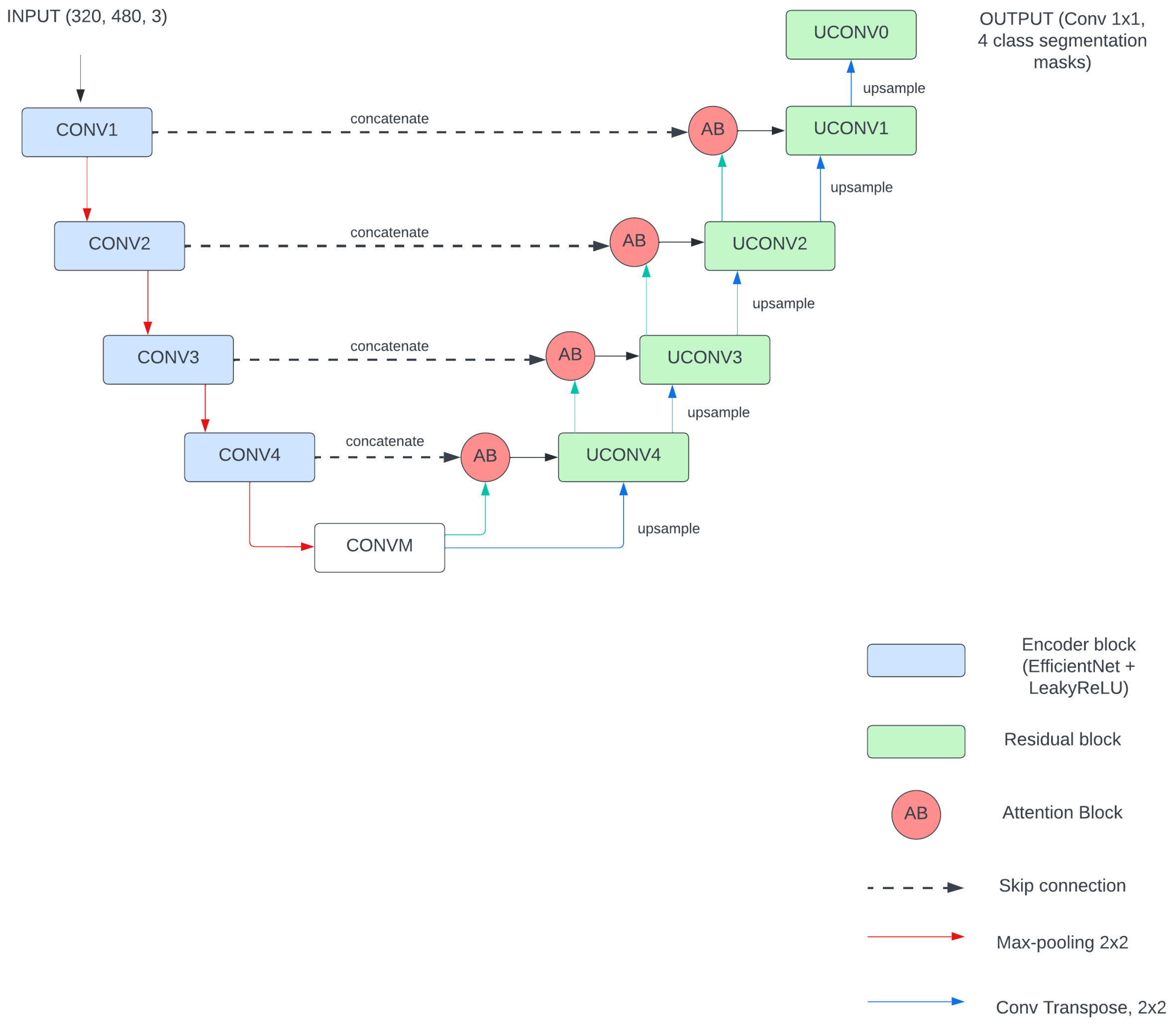


**Figure 5.** Proposed model architecture.

### *5.2. Training and Testing*

1. Custom data generators were created for training and validation with a batch size of 8 and the image input size of 320 × 480. This batch size and input size was chosen in particular due to the CPU memory constraints.
2. During training, two different loss functions were implemented: binary cross-entropy (BCE) and Dice loss [27,28].
3. A combination(sum) of these was used, which gives clear boundaries and works much better than a single loss function individually.
4. The model was trained for 30 epochs. As a pre-trained encoder was used, the model tended to overfit beyond 30 epochs, thereby showing no significant improvement in the scores.

#### 5.2.1. Dice Loss

Dice loss gives us an idea of how close the predicted map is to the ground truth. It is calculated as

$$DL(y, p) = 1 - \frac{[2 \times p \times y] + 1}{y + p + 1} \tag{1}$$

where $y$ is the actual value $p$ is the predicted value.

5.2.2. Binary Cross-Entropy

$$BCE(y, p) = -(y \log(p) + (1 - y) \log(1 - p)) \tag{2}$$

where $y$ is the actual value, and $p$ is the predicted value.

5.2.3. Optimizer

NAdam optimizer with a learning rate of 0.002, an improvement of Adam, was applied for faster convergence and memory efficiency [29].

## 6. Results

### *6.1. Evaluation Metrics*

IoU, F1 score, Dice coefficient, and Dice loss (for validation) are various metrics used for the model evaluation. The Dice coefficient differs from the F1 score value for multiclass segmentation of images. Hence, both metrics must be evaluated [30].

6.1.1. Intersection-Over-Union (IoU) [31]

IoU (also known as Jaccard index) is a commonly used metric to measure the overlap between two masks, especially for segmentation. $IoU = Areaofoverlap/Areaofunion$ The IoU score is close to 1 if the predicted masks match the ground truths and approach 0 with a decrease in the overlap.

6.1.2. F1 Score

By calculating the harmonic mean of a classifier's accuracy and recall, the F1-score integrates both into a single statistic. The $F1_{Score}$ of a classification model is calculated as follows:

$$F1_{Score} = \frac{2 \times P \times R}{P + R} \tag{3}$$

$P$ = The precision of the classification model $R$ = The recall of the classification model.

6.1.3. Dice Coefficient

This is like precision, as it evaluates the score and penalizes for wrong pixel classification. It is given by

$$DC = 2 \times \frac{Area\ of\ overlap}{Total\ number\ of\ pixels\ in\ (A + B)} \tag{4}$$

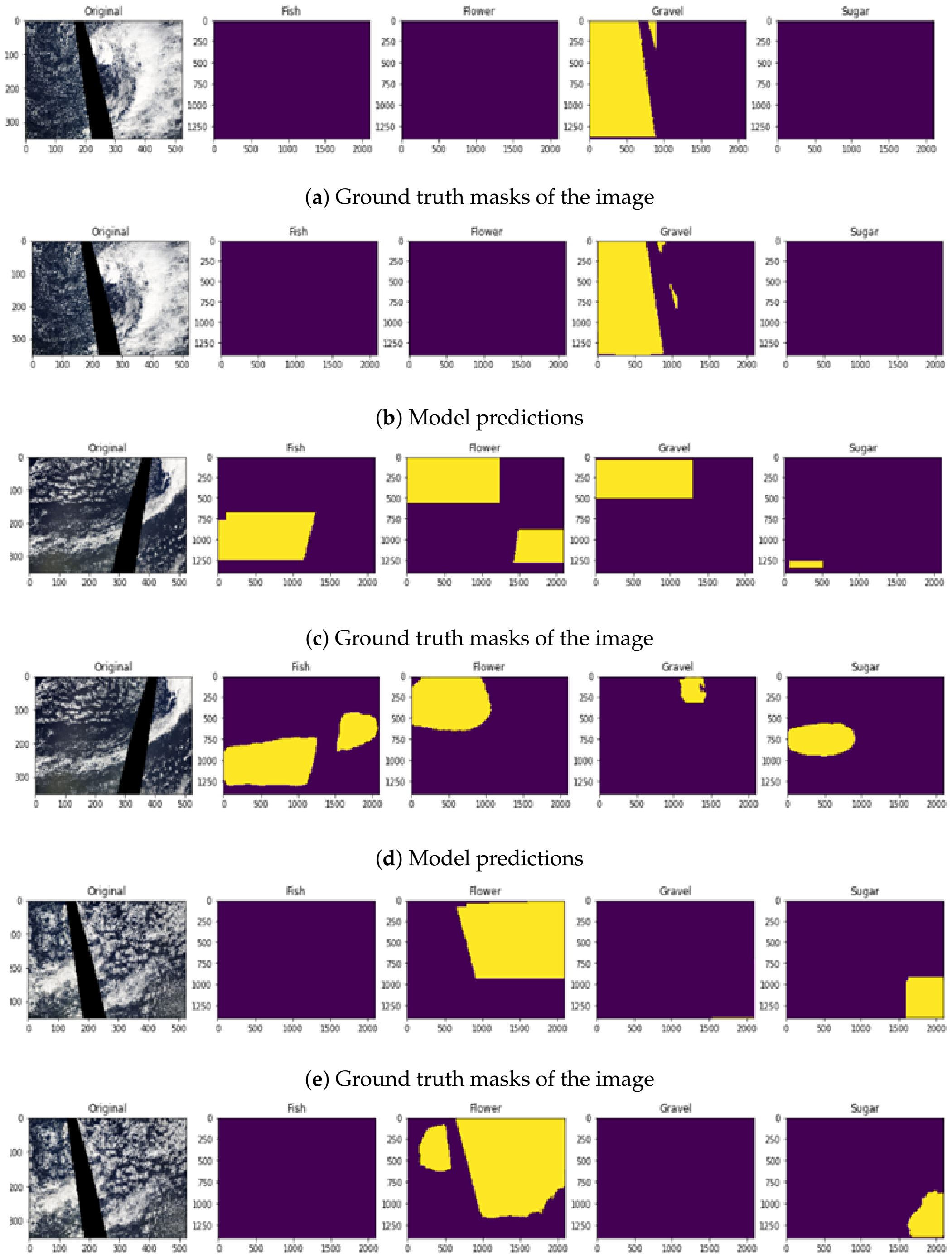
where $A$ and $B$ are the predicted mask and ground truth, respectively. The output comparison of the masks is shown in Figure 6 below.

### *6.2. Output Comparison*

The most widely used SOTA model for segmentation tasks is the UNet with Resnet34 as a pre-trained encoder (ResUNet). The proposed model outperformed in the four metrics, as shown in Table 3, and proved an improvement of more than 2% in each.

**Table 3.** Results.

| Model Used | Validation Loss | IoU | Dice Coefficient | F1 Score |
|---|---|---|---|---|
| ResUNet | **0.7639** | 0.4078 | 0.5437 | 0.5553 |
| EfficientNet encoder | 0.7580 | 0.4227 | 0.5537 | 0.5733 |
| Efficient net encoder + residual blocks | 0.7535 | 0.4147 | 0.5504 | 0.5629 |
| Our Model | 0.7424 | **0.4239** | **0.5557** | **0.5735** |
| Improvement w.r.t. ResUNet | −2.81% | +3.95% | +2.2% | +3.28% |



(**a**) Ground truth masks of the image

(**b**) Model predictions

(**c**) Ground truth masks of the image

(**d**) Model predictions

(**e**) Ground truth masks of the image

(**f**) Model predictions

**Figure 6.** Results of 3 images with their ground truths and predicted masks respectively.

## 7. Conclusions

Deep learning techniques have their own set of pros and cons [32,33]. By using various deep learning techniques in ensemble architecture, we produced a method to recognize complex cloud patterns. The proposed model also outperforms the present state-of-the-art techniques used for cloud pattern recognition, thus enhancing accuracy. After applying various preprocessing and post-processing techniques, the UNet-based encoder–decoder architecture reached a F1 score of 0.5735. This area has immense potential for further research, and our approach tends to serve as a benchmark for future research. Our future work will focus on improving the SOTA techniques' ensemble by incorporating the advantages of each. Further, we believe that the addition of [34] atrous spatial pyramid pooling (ASPP) modules could help upgrade the existing architecture.

**Author Contributions:** Conceptualization, S.N., K.S. (Kunal Shah) and R.V.B.; formal analysis, B.Z. and R.V.B.; investigation, S.N. and K.S. (Kunal Shah); methodology, S.N. and K.S. (Kunal Shah); project administration, B.Z., R.V.B., D.M., K.S. (Kailash Shaw) and V.J.; software, S.N. and K.S. (Kunal Shah); supervision, B.Z., R.V.B., D.M., K.S. (Kailash Shaw) and V.J.; validation, B.Z., R.V.B., D.M., K.S. (Kailash Shaw) and V.J.; visualization, B.Z. and R.V.B.; writing—original draft, S.N. and K.S. (Kunal Shah); writing—review and editing, B.Z., R.V.B., D.M., K.S. (Kailash Shaw), V.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Research Support Fund (RSF) of Symbiosis International (Deemed University), Pune, India.

**Data Availability Statement:** Not applicable.

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

## References

1. Arking, A. The Radiative Effects of Clouds and their Impact on Climate. *Bull. Am. Meteorol. Soc.* **1991**, *72*, 795–814. [CrossRef]
2. Song, X.; Liu, Z.; Zhao, Y. Cloud detection and analysis of MODIS image. In Proceedings of the 2004 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2004), Anchorage, AK, USA, 20–24 September 2004; Volume 4, pp. 2764–2767.
3. Mahajan, S.; Fataniya, B. Cloud detection methodologies: Variants and development—A review. *Complex Intell. Syst.* **2019**, *6*, 251–261. [CrossRef]
4. Audebert, N.; Le Saux, B.; Lefèvre, S. Segment-before-Detect: Vehicle Detection and Classification through Semantic Segmentation of Aerial Images. *Remote Sens.* **2017**, *9*, 368. [CrossRef]
5. Henry, C.; Azimi, S.M.; Merkle, N. Road Segmentation in SAR Satellite Images With Deep Fully Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2018**, *15*, 1867–1871. [CrossRef]
6. Arbelaez, P.; Hariharan, B.; Gu, C.; Gupta, S.; Bourdev, L.; Malik, J. Semantic segmentation using regions and parts. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012), Providence, RI, USA, 16–21 June 2012; pp. 3378–3385. [CrossRef]
7. Shanmugam, D.; Blalock, D.; Balakrishnan, G.; Guttag, J. Better Aggregation in Test-Time Augmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 1214–1223.
8. Mane, D.; Bidwe, R.; Zope, B.; Ranjan, N. Traffic Density Classification for Multiclass Vehicles Using Customized Convolutional Neural Network for Smart City. In *Communication and Intelligent Systems*; Sharma, H., Shrivastava, V., Kumari Bharti, K., Wang, L., Eds.; Springer Nature: Singapore, 2022; pp. 1015–1030.
9. Brezočnik, L.; Fister, I.; Podgorelec, V. Swarm Intelligence Algorithms for Feature Selection: A Review. *Appl. Sci.* **2018**, *8*, 1521. [CrossRef]
10. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
11. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
12. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; Chaudhuri, K., Salakhutdinov, R., Eds.; Volume 97, pp. 6105–6114.
13. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems* ; MIT Press: Cambridge, MA, USA, 2017; Volume 30.
14. Oktay, O.; Schlemper, J.; Folgoc, L.L.; Lee, M.C.H.; Heinrich, M.P.; Misawa, K.; Mori, K.; McDonagh, S.G.; Hammerla, N.Y.; Kainz, B.; et al. Attention U-Net: Learning Where to Look for the Pancreas. *arXiv* **2018**, arXiv:1804.03999.
15. Baheti, B.; Innani, S.; Gajre, S.S.; Talbar, S.N. Eff-UNet: A Novel Architecture for Semantic Segmentation in Unstructured Environment. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Virtual, 14–19 June 2020; pp. 1473–1481.
16. Li, D.; Dharmawan, D.A.; Ng, B.P.; Rahardja, S. Residual U-Net for Retinal Vessel Segmentation. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1425–1429.
17. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
18. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]

19. Kalinin, A.A.; Iglovikov, V.I.; Rakhlin, A.; Shvets, A.A. Medical Image Segmentation Using Deep Neural Networks with Pre-trained Encoders. In *Deep Learning Applications*; Wani, M.A., Kantardzic, M., Sayed-Mouchaweh, M., Eds.; Springer: Singapore, 2020; pp. 39–52. [CrossRef]
20. Bae, M.H.; Pan, R.; Wu, T.; Badea, A. Automated segmentation of mouse brain images using extended MRF. *NeuroImage* **2009**, *46*, 717–725. [CrossRef] [PubMed]
21. Understanding Clouds from Satellite Images Crowd-Sourcing Activity. Available online: https://www.zooniverse.org/projects/raspstephan/sugar-flower-fish-or-gravel (accessed on 7 October 2022).
22. Buslaev, A.; Iglovikov, V.I.; Khvedchenya, E.; Parinov, A.; Druzhinin, M.; Kalinin, A.A. Albumentations: Fast and Flexible Image Augmentations. *Information* **2020**, *11*, 125. [CrossRef]
23. Wang, G.; Li, W.; Ourselin, S.; Vercauteren, T. Automatic Brain Tumor Segmentation using Convolutional Neural Networks with Test-Time Augmentation. *arXiv* **2018**, arXiv:1810.07884.
24. Lee, J.; Won, T.; Hong, K. Compounding the Performance Improvements of Assembled Techniques in a Convolutional Neural Network. *arXiv* **2020**, arXiv:2001.06268.
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778. [CrossRef]
26. Chen, X.; Yao, L.; Zhang, Y. Residual Attention U-Net for Automated Multi-Class Segmentation of COVID-19 Chest CT Images. *arXiv* **2020**, arXiv:2004.05645.
27. Jadon, S. A survey of loss functions for semantic segmentation. In Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), Virtual, 27–29 October 2020; pp. 1–7. [CrossRef]
28. Moltz, J.H.; Hänsch, A.; Lassen-Schmidt, B.; Haas, B.; Genghi, A.; Schreier, J.; Morgas, T.; Klein, J. Learning a Loss Function for Segmentation: A Feasibility Study. In Proceedings of the 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), Iowa City, IA, USA, 3–7 April 2020; pp. 357–360.
29. Dozat, T. Incorporating Nesterov Momentum into Adam. In Proceedings of the ICLR Workshop, San Juan, PR, USA, 2–4 May 2016.
30. Wang, Z.; Wang, E.; Zhu, Y. Image Segmentation Evaluation: A Survey of Methods. *Artif. Intell. Rev.* **2020**, *53*, 5637–5674. [CrossRef]
31. van Beers, F.; Lindström, A.; Okafor, E.; Wiering, M. Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, Prague, Czech Republic, 19–21 February 2019; Volume 1, pp. 438–445. [CrossRef]
32. Bidwe, R.V.; Mishra, S.; Patil, S.; Shaw, K.; Vora, D.R.; Kotecha, K.; Zope, B. Deep Learning Approaches for Video Compression: A Bibliometric Analysis. *Big Data Cogn. Comput.* **2022**, *6*, 44. [CrossRef]
33. Zope, B.; Mishra, S.; Shaw, K.; Vora, D.R.; Kotecha, K.; Bidwe, R.V. Question Answer System: A State-of-Art Representation of Quantitative and Qualitative Analysis. *Big Data Cogn. Comput.* **2022**, *6*, 109. [CrossRef]
34. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.