Squish: A Smooth Self-Regularizing Activation Function

Christian Zamiela, Durant Fullington, Preston Robertson,
Industrial and Systems Engineering Department,
Mississippi State University,
Mississippi State, 39762

Abstract

Deep learning has become an extremely important tool for a wide range of data analysis applications and the choice of activation function is a critical design choice for optimizing model performance. Many activation functions lack control over the upper bound, which can often lead to exploding gradient problems. We propose a novel Squish activation function with a self-regularizing lower bound and controlled upper bound. The Squish activation function builds upon the self-regularizing lower bound of Swish function by merging (*'Squishing'*) the SoftSign function with parameter K for control over the upper bound. The proposed function increases the variance for improved low-level feature learning and posses a smooth landscape for improving model performance with random weight initialization. Our experiments show that Squish tends to work better than alternative activation functions across a diverse set of image segmentation data sets. The Squish performance is validated by comparing the mean intersection over union metric (M-IOU) score, where it achieve the highest average score of 0.7127.

Keywords: Activation Function; Deep Learning; Image Segmentation

1 Introduction

1.1 Deep Learning and Activation Functions

Deep Learning (DL) models have found great success within a variety of diverse applications, including natural language processing, visual data processing, and audio processing [1]. Within these different domains, one of the more prominent applications of DL is for processing visual image data. This includes tasks such as image classification [2, 3], object detection [4, 5, 6], image segmentation [7, 8], and video processing [9]. The great success of DL models within these different applications can be attributed to the activation function [10].

The activation function plays a crucial role in the network, by taking an input value from the previous layer, combined with the weight and bias vectors, and returns an output

activation value that is passed to the following layer. Activation functions are leveraged for their ability to add non-linearity to the network. This non-linearity can be attributed to the success of DL models for learning complex, non-linear patterns [11]. Without this added non-linearity, the output from each layer within the network would essentially be a simple linear function [12], leading to an inability to capture complex patterns present in the data. This is especially crucial for processing image data, where the data is highly complex and non-linear. The motivations for this research are proposed and discussed by *Ranjan et al.*, as the activation functions are still a developing feature of DL models and there is still ample opportunities for novel improvements [12].

In addition to adding non-linearity to the network, there are several other important important properties that an activation function should posses [12]. Firstly, the activation function must be differentiable to successfully compute back propagation error and gradient descent loss for training the model. In addition, there needs to exist a gradient, where the value is greater than or equal to one, but still a relatively small value, for improved learning of low-level features. This gradient must also avoid common learning challenges, such as vanishing gradients (small gradients) [13] and exploding gradients (large gradients). Furthermore, there needs to exist a saturation region, where the gradient is equal to zero. This is a desired property as it leads to a reduction in variance [12]. Finally, two additional properties that are desirable for activation functions are normalization and regularization characteristics. which help the network to learn from the training data without suffering from challenges relating to overfitting.

1.2 Squish Activation Function

This research proposes to further improve the performance of DL models, specifically within the realm of Image Segmentation. Our proposed activation function, Squish, is design to combine the valuable portions of both the Swish and the SoftSign activation functions, developing a stronger and better performing activation function (visualized in Figure 1). The design and objectives of this activation function are based on the motivations and desired activation function properties discussed in [12].

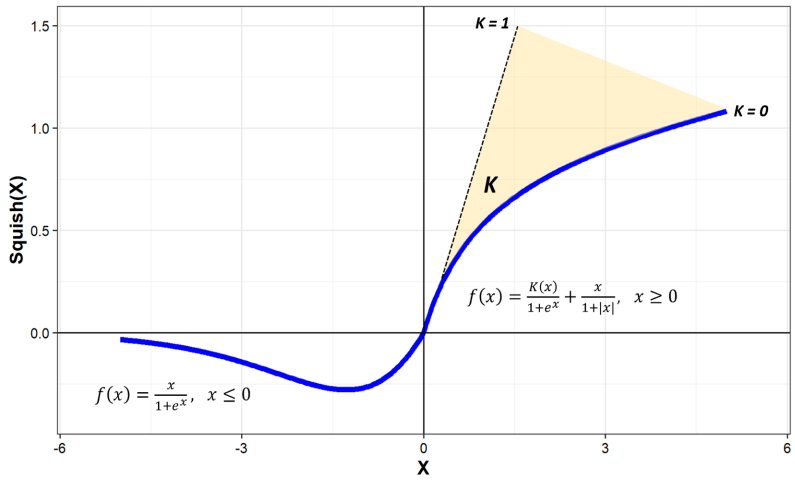

Figure 1: Novel Squish activation function, where the balance between bounded and unbounded function is controllable by parameter K that ranges from 0-1.

Firstly, our activation function is **nonlinear**, **non-monotonic**, and **smooth**. These are essential properties, as they help the network to converge faster and capture the extensive nonlinearities in image segmentation data. In addition, our proposed activation function has a novel bound-to-unbounded control parameter, K. This parameter allows the user to tune the slope in the activation function for values > 0. This advantage leads to enhanced performance and better generalization for the DL network, as the user can tune the network activation functions to better fit the applications, with respect to the amount of bounded-to-unbounded desired. Furthermore, our proposed activation function has two additional properties, including **self-regularization** and **fixed larger gradients**. These two additional properties help to combat both overfitting and exploding gradient problems, which are both common challenges in DL networks. Overall, the contribution of our proposed activation function can be summarized as follows:

1. Enhanced noise reduction due to self-regularizing lower bound and controlled gradual slope for improved structural detection.

2. A fixed larger gradient for robust and generalize learning of lower-level features that alleviates the exploding gradient problem.

3. Improved training with random weight initialization due to smooth landscape of activation function to prevent vanishing gradient.

4. Provide a methodology for combining activation functions for enhanced non-linear mapping of complex features.

The proposed Squish function is validated with experimental data and a case study with image segmentation data. Firstly, the proposed activation function will be compared with several competitive alternatives (ReLU, ELU, Swish) using synthesized data. This will allow for a better understanding of the performance of each activation function for feature extraction, within a controlled setting. Secondly, five (5) diverse data set will be leveraged to gauge the application performance of the proposed activation function. These different data set will showcase both the improved performance and the overall generalizability of the Squish activation function.

The remaining paper is organized as follows. Section 2 will discuss the theoretical and mathematical properties of the proposed activation function. Section 3 will cover the experimental validation of the proposed activation function against several comparable and widely used activation functions. Section 4 will provide insight into the real data analysis case study, including details about the data sets, experimental procedures, and performance comparison and evaluation. Section 5 will provide a discussion on the results from section 4, and section 6 will conclude the paper.

2 Proposed Squish Activation Function

We propose a Squish activation function with self-regularizing lower bound and controlled upper bound. The Squish function combines the SoftSign and Swish functions with a control parameter K. The essential properties in Squish activation function that are discussed by *Ranjan et al.* are 1) non-linearity, 2) a fixed larger gradient for enhance learning of low-level features but we control the variance a fixed point prevent explosion, 3) saturation region for reduced variance, and 4) continuously differentiable for optimization and data set generalization. Furthermore, three theoretical properties are discussed in light of the essential properties in this section, 1) smoothing landscape, 2) self-regularizing, and 3) controlled gradient. The listed properties are found in the Squish activation function, which is defined as:

$$f(x) = \begin{cases} \frac{x}{1+e^x} & x < 0 \\ \frac{Kx}{1+e^x} + \frac{x}{1+|x|} & x \geq 0 \end{cases} \tag{1}$$

where K is a trainable and tuneable parameter that is leveraged to control the upper bound. For this specific research, the Swish and SoftSign functions are combined to achieve balance with the trade offs of saturation near zero, to reduce variance for lower bound, and having a gradual slope for the upper bound. The SoftSign was chosen, over Sigmoid or Tanh, due to the more gradual slope of the activation function. The unbounded region allows the neural network to learn lower level features that can be overlooked with bounded functions. Another important aspect of the activation function is the derivative, where the Squish activation function is defined as:

$$f'(x) = \begin{cases} \frac{e^{-x}(x+1)+1}{(1+e^{-x})^2} & x < 0 \\ \frac{K(e^{-x}(x+1)+1)}{(1+e^{-x})^2} + \frac{x}{(1+|x|)^2} & x \geq 0 \end{cases} \tag{2}$$

The derivative of proposed activation function can be visualized in Figure 2. The novelty of this activation function comes from the combination of different functions to achieve the desired theoretical properties. The thresh-holding parameter, K, controls the slope of upper bound by combining the positive regions of the swish function with SoftSign function. In Figure 1, we observe the range of K and the boundaries are $[\approx -0.27, \infty)$ when $K > 0$. K ranges between $0 - 1$ and is both trainable and tune-able. However, training adds another weight that leads to an increase in the computational complexity.

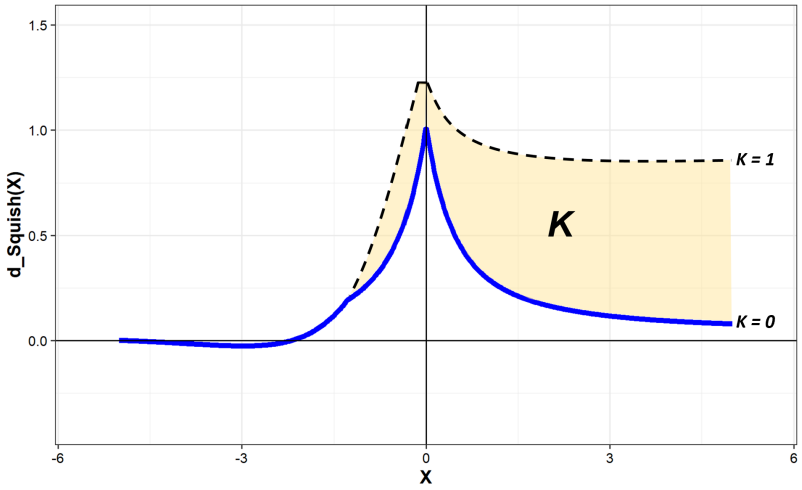

Figure 2: Derivative of novel Squish activation function, where the balance between bounded and unbounded function is controllable by parameter K

2.1 Theoretical properties

2.1.1 Property 1: Smoothing Landscape

Over saturation can occur when value get too close to bounded value that it can no longer escape that point after iterations of training. Saturation causes the weights during training to be too small, slowing and preventing convergence of DL models. The Sigmoid and Tanh functions commonly experience this problem because the functions saturate in the upper and lower bound. Vanishing gradient is a common problem associated with saturation. The Relu function is a non-negative function that causes vanishing gradient problems due to the derivative possessing limited training capabilities. It is valuable for the derivative to range between a set values, rather than be binary values of zero or one. A gradient implemented with swish lower bounded mitigates the vanishing gradient problem. Activation function such as Elu and Leaky-Relu proceed below zero to prevent vanishing gradient and improve learning of lower level features. Lastly, a smoother transition makes a DL neural network less sensitive to initialization weights, improving consistency of results [14]. Smoother loss landscapes are easier to optimize and result in better training and test accuracy. The smoothing landscape of Squish activation function can be visualized in Section 3.3.

2.1.2 Property 2: Self Regularizing

Over fitting to features and noise within a data set is challenge for many DL model. However, regularizing removes noise and allows for better convergences of loss function. The swish function is a self regularizing function due to the non-monotonic nature of lower bound approaching ≈ -0.27 and proceeding back toward zero over many training iterations, increasing variance and lower-level learning. As the values approach saturation at zero from the lower bound the variance is dampen if its too large in the lower layers [11]. A self regularizing effect can be visualized in Figure 1, from the proposed Squish function, which employs a swish like curve for negative values. Lastly, the slope of Squish function being > 1 in the early stages of training provides more learnable values due the increased variance. After many iterations of training the number of learnable values decreases due to the regularizing affect of curved upper slope in Squish function.

2.1.3 Property 3: Controlled Gradient

A small range of gradient values results in the variance between values to be smaller in lower layers, where low-level features are learned. However, a large gradient can lead to exploding and vanishing gradient problems. This is because the model will over learn these low-level features, creating noise and leading to slow convergence problems. Controlling the gradient during backprogration learning leads reduced noise and lower-level feature learning. A gradient > 1 and $< 1+\delta$ increases the variance but a small δ prevents exploding and vanishing gradient [12]. The controlled gradient of the proposed Squish function can be visualized in Figure 2 highlighting the added flexibility from the novel parameter, K. Increasing K allows Swish function to have a larger impact on positive values, increases the size of the gradient.

3 Experimental Validation

This section will detail the preliminary experimental validation, which will provide useful insight into the performance of the proposed Squish function against several commonly used and competitive activation functions. This experimentation is conducted on synthesized data, which produces a controllable and comparable environment for evaluation. The

following subsections will include the other common activation functions comparable to the Squish, activation non-linear learning estimation, and a loss landscape estimation.

3.1 Common Activation Functions

Many activation function have been widely used in literature. Some of the following activation functions will be used as a comparison metric when evaluating the proposed novel activation function. Below we list common activation functions and describe important features [10, 15, 16].

1. Rectified Linear Unit (ReLU)

$$f(x) = max(0, x) \tag{3}$$

   Most common activation function due to simple gradient and fast convergence.

2. Leaky Rectified Linear Unit (Leaky ReLU)

$$f(x) = \begin{cases} \alpha x & x < 0 \\ x & x \geq 0 \end{cases} \tag{4}$$

   Introduces non-zero gradient to avoid vanishing gradient. Where α removes the lower boundary of Relu activation function.

3. Exponential Linear Unit (ELU)

$$f(x) = \begin{cases} e^x - 1 & x < 0 \\ x & x \geq 0 \end{cases} \tag{5}$$

   Provides an exponential smoothing transition between positive and negative values and saturates slighly below zero.

4. SeLU

$$f(x) = \lambda \begin{cases} \alpha(e^x - 1) & x < 0 \\ x & x \geq 0 \end{cases} \tag{6}$$

   Self-normalizing function that push values closer to zero and provides consistent variance. Where λ is a scaling parameter and α is a smoothing parameter.

5. Swish

$$f(x) = \frac{x}{1 + e^x} \tag{7}$$

Adaptation of sigmoid function that is self-regularizing due to non-monotonic property that cause values to reach boundary just below zero and trend back toward zero to saturate.

3.2 Non-linear Learning Estimation with Gaussian Kernel Density Probability Distribution

Three different distributions are generated to evaluate the performances of our proposed activation function. These distributions are generated using a Gaussian kernel density estimation. This provides a probability density function of the randomly generated Gaussian data points, re-scaled to be between (-1,1) to better capture the effective range of each activation function. This experimental set-up is a similar tactic as the one used in [17], where the probability density function is used to evaluate how well the activation function can capture patterns and nonlinearities in the data. Overall, this experimentation highlights and validates the main theoretical advantages of the Squish function.

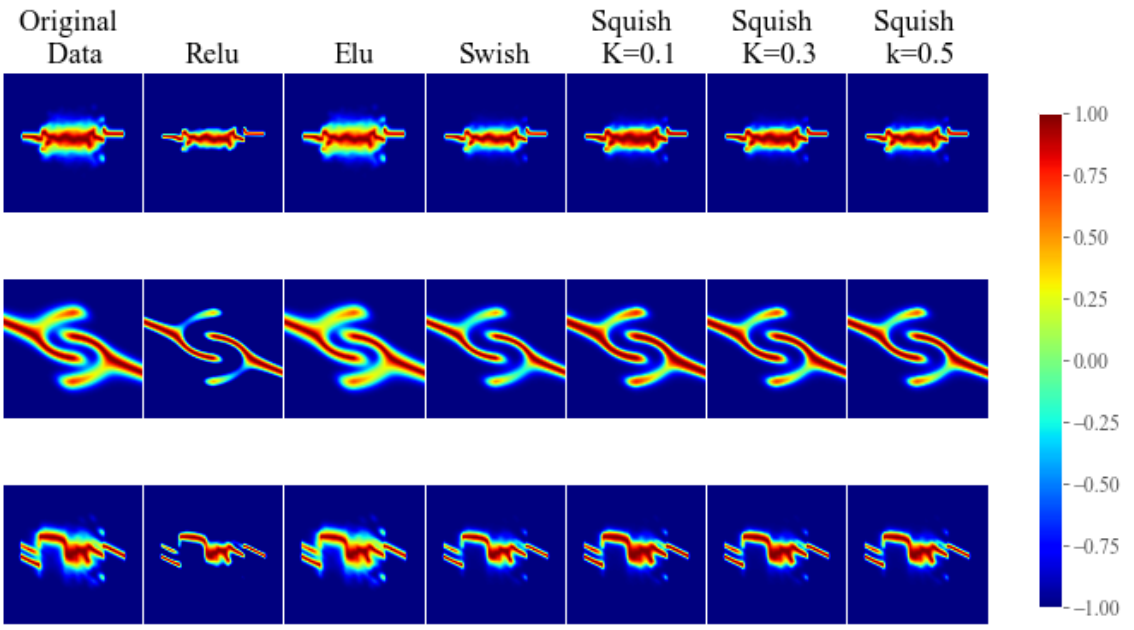

Figure 3: Gaussian kernel density probability distribution transformed through activation functions. The color scale represents the magnitude of the density function, between -1 and 1

Figure 3 and Figure 4 represent a qualitative approach to gauging the experimental performance of the the Squish function, compared to several other commonly used functions. To start, Figure 3 details how the Squish function is able to extract a high level of features from the Gaussian distributions, comparatively or better than all other functions. This is observed through the sharper details captured through the Squish functions ($K = 0.1, 0.3, 0.5$), especially as the ReLU and Swish functions capture arguably less details. In addition, compared with the ELU function, the Swish function creates sharper boundaries around the distribution features. This causes the Squish function to potentially miss some of the less extreme values, but it also highlights its ability to create sharp boundaries to discriminate the distribution from the background.

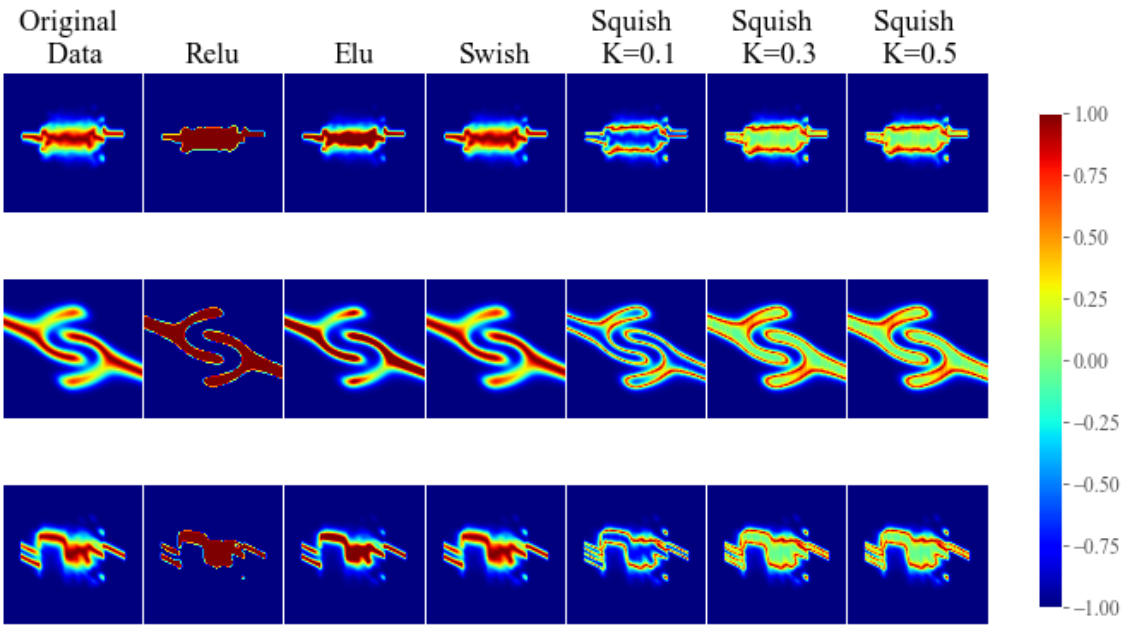

Figure 4: Gaussian kernel density probability distribution features learned through activation function derivatives. The color scale represents the magnitude of the density function, between -1 and 1

Furthermore, in Figure 4, the derivatives of the different activation functions are evaluated. The derivative is an important evaluation metric due to its role in the process of backpropagation, where the actual model training occurs. The Squish function is shown to behave differently than the other commonly used functions, where it provides more emphasis on the edge features of the distribution. These are the features of the distribution that are directly correlated with the structural alignment of the object. The Squish function's ability to distinctively highlight these boundaries provides an unique advantage for image segmentation applications. In addition, the Squish function is more sensitive to various features, as opposed to the Relu function which is learning all feature evenly, as depicted in Figure 4. The edge features are considered to be lower-level features, and the Squish's sensitivity to these features allows the it to better identify complex shapes during training.

3.3 Loss Landscape Estimation

The problem with optimizing neural networks is minimizing the loss. Optimizers guide the network toward the lowest point from randomly initialized weights during training. The success of training across a surface is dependent upon the smoothness and shape of the landscape. In Figure 5, the loss sensitivity of a randomly generated 5-layer neural network is evaluated based on the proposed Squish function, compared to other commonly used functions. This 5-layer neural network is compiled using mean squared error loss function and Adam optimizer to fit to a parabola curve [18]. From Figure 5, it is observed that the Elu and Swish functions have several sharp transitions, as compared to the Squish (K=0.5). In addition, the Sigmoid and Squish ($K = 0.0$) are bounded functions, which have smaller convergence locations. However, the Squish ($K = 0.1, 0.3, 0.5$) upper values are unbounded. The Squish ($K = 0.1$) is characterized by a shallow slope region, likely leading to smaller convergence location. However, when the Squish parameter K is equal to 0.3 or 0.5, there is a smoother and more uniform transition toward the lower loss values. This smoother change in loss values leads to more consistent convergence with randomly initialized weights.

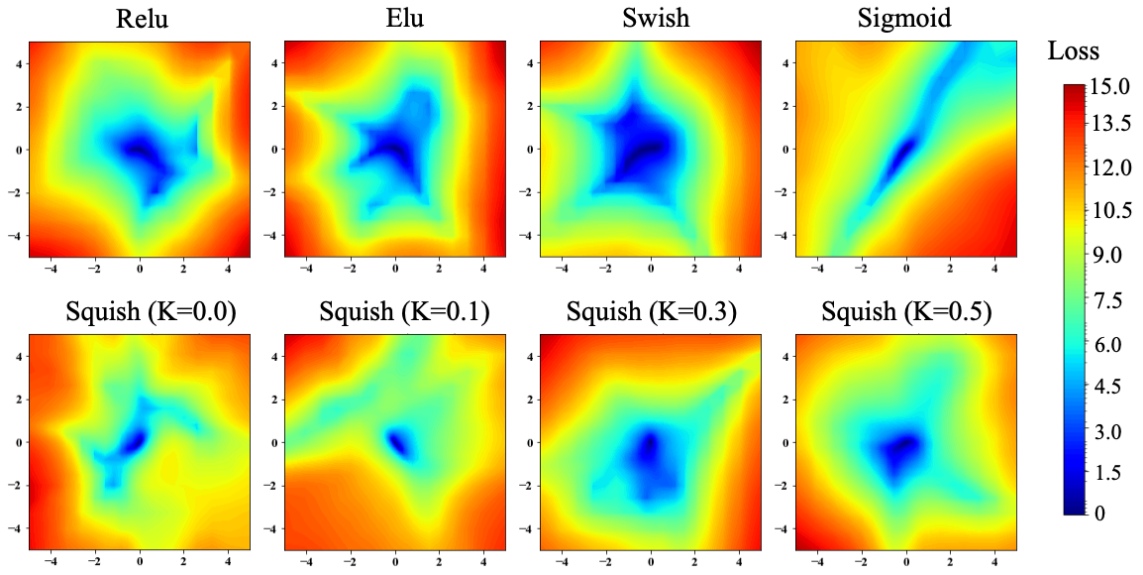

Figure 5: Loss landscape of 5-layer neural network tested on several activation functions.

4 Case Study: Squish Out Preforms Alternative Activation Functions for Image Segmentation with U-Net Deep Learning Model

These experiments, and their evaluations, aim to show the value of the proposed activation function and enhanced edge detection for image segmentation applications. First, the combination parameter K is tuned and trained for each data set. This parameter controls the upper bound of the activation function, and in turn, allows the model to control training. Training happens by allowing the back propagation in neural network to select parameter K. Second, the proposed activation function is then compared to the other popular DL activation functions. Gauging the Squish functions performance in a deep leaning network shows the value the proposed functions properties over alternative options.

Training Settings: The U-Net is image segmentation DL neural network that has four encoder and decoder blocks with convolutional layers for image processing. The model begins with 16 neurons in the first encoder block and double the number of neurons after each encoder block, until reaching the latent space. The latent space is then up sampled with the decoder block, and the number of neurons is reduced by half after each block. From there, dropout layers and skip connections are used to provide a regularizing effect in each block. The model is trained using 200 epochs, binary cross-entropy loss function, and Adam optimizer. Model training is takes place using high performance computing GPU systems

Evaluation Metrics: The four selected metrics for model validation are (1) Mean Intersection of Union (Mean IoU), or Jaccard's Index, which measures the percent overlap between the predicted segmentation map and the ground truth segmentation map [19, 20]. (2) Peak signal to noise ratio (PSNR), which measures noise through the efficiency of compressors [21]. Where the ratio between the maximum power of an image and the maximum power of the noise corrupting the image is captured, and outputs a scalar value to represent the quality of the segmentation map. (3) F1 score (F1) is the balance the strength between between recall and precision to observe the number of correct predictions and highlight some of the poorer performing instances [22]. (4) Pixel accuracy (PA) is number of correct pixel predictions over the total number of predictions and is the foundation metric

to many of the segmentation metrics.

4.1 Characteristics of Data Sets

Performance of the proposed Squish activation function is verified using five data sets for binary image segmentation. These five data sets include a diverse set of images, spanning from animals, humans, water bodies, flowers, and brain MRI scans. This selection of data sets is simulating the use of the proposed activation function within a wide variety of applications, containing irregular and complex shapes within the images. This provides a more effective and robust method of gauging the Squish performance. Furthermore, each data set contains both the original image and an accurate ground truth label that is used to validate

Before leveraging these data sets in experimentation, each underwent a separate pre-processing procedure to fit the image to the U-Net model. These procedures included, re-scaling the input images to 256×256 resolution, converting to gray scale (one-channel), and removing distorted, noisy, or unbalanced images from the data set. In addition, some of the data sets contained multi-class segmentation maps (Oxford Pets, Oxford Flowers, and Pascal People), which required an additional preprocessing step to convert these maps into binary segmentation maps, focusing on only one object class within the images. The following sections will briefly provide insight into each data set and sample image sets from each data set can be seen in Figure 6.

Pascal People: The Pascal People data set represents a sub-sampled image set from the Pascal Visual Object Challenge (VOC) 2012 data set [23]. These images were originally contained multi-class ground truth segmentation maps, which containing 19 different classes other than 'human'. During preprocessing these multi-class segmentation maps were converted into single-class maps, representing only the human class. In addition, the images of people are not only static, but contain images of 'People in action', which adds complexity to the data set. From the available 5,826 images containing human classes, 546 images were leveraged for evaluating the proposed activation function after preprocessing.

Brain MRI: The Brain MRI data set contains a set of patient brain scans, which corresponds to either healthy brain scans or the presence of abnormalities. These images were originally apart of the the Cancer Imaging Archive, and have since been made publicly

available [24]. From this data set, 1,000 images were leveraged for evaluating the proposed activation function.

Water Bodies: The Water-Bodies Data set consists of a variety of different satellite images of rivers, lakes, and oceans captured from high altitudes [25]. In total, there are 2,841 different images, each containing one specific body of water. These images are extremely complex, with fine details corresponding to the different branches of the rivers and deltas and complex shorelines. From this image set, after preprocessing and removing noisy and distorted images, 1,000 images were leveraged for evaluating the proposed activation function.

Oxford Flowers: The Oxford Flowers data set is a large data set containing 1,360 images of different species of flowers [26]. There are 17 different classes, each class having 80 represented images. The different flower petals have a variety of complex geometries and intricate details that provide an excellent evaluation of different activation functions. From this data set, 1,000 images were sub-sampled for evaluating our proposed activation function against other common activation functions.

Oxford Pets: The Oxford Pets data set is a large data set containing roughly 7,500 images of different domesticated pets (Cats and Dogs) [27]. This data set contains 37 different breeds of dogs and cats, where each breed of pet (category) has approximately 200 representative images. These images contain a large amount of complex and sparse features, such as hair, ears, and tails. From this set of images, 1,000 were sub-sampled for evaluating our proposed activation function and other commonly used activation functions.

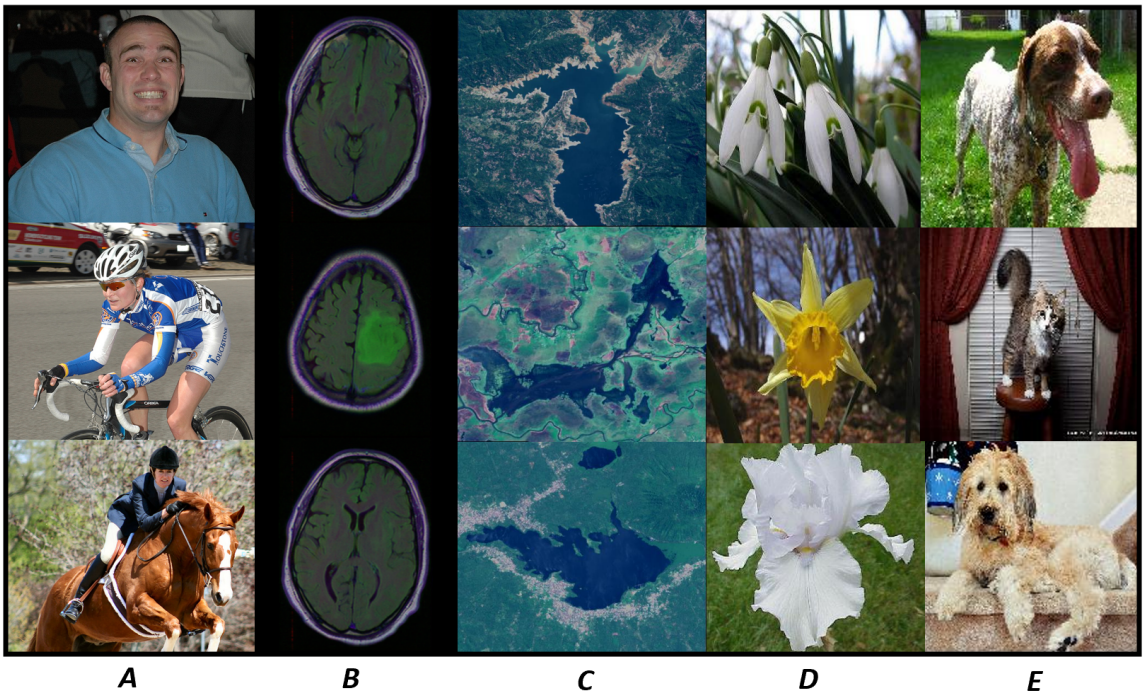

Figure 6: Example of images from data sets Pascal People (A), Brain MRI (B), Water Bodies (C), Flowers (D), and Oxford Pets (E)

4.2 Compare Tuned and Trained Squish Parameter K

First, we compare the Squish function with different tuning values for K, over the five image segmentation data sets. In order to evaluate the effect of each K value, the mean intersection over union (M-IOU) score is leveraged as a quantitative evaluation metric. For this evaluation, it is observed that as K increases, the variance of values fed through activation function increases. The understanding of the interaction between K and variance is necessary to determine the best parameter for achieving the highest overlap between y_{true} and y_{pred} segmentation maps. From Section 3.3, increasing parameter K will result in a smoother convergence landscape and better M-IOU score. In Table 1, it is determined that the best value of K each data set is not consistent in order to achieve the highest mean intersection over union score, however, the most common preference is $K = 0.3$.

Table 1: Squish M-IOU scores for five testing data sets and different values of K.

	Pascal People	Brain MRI	Water Bodies	Oxford Flowers	Oxford Pets	Average M-IOU
Squish (K=0)	**0.4801**	0.6758	0.7397	0.8642	0.7794	0.7078
Squish (K=0.1)	0.4754	0.6988	0.7442	0.8062	0.7805	0.7010
Squish (K=0.2)	0.4622	0.6970	0.7490	0.8617	0.7725	0.7085
Squish (K=0.3)	0.4682	0.6948	**0.7491**	**0.8648**	**0.7867**	**0.7127**
Squish (K=0.4)	0.4403	**0.7039**	0.7399	0.8636	0.7769	0.7049
Squish (K=0.5)	0.4474	0.6898	0.7430	0.8642	0.7782	0.7045
Squish (K= Train)	0.4653	0.6607	0.7369	0.8568	0.7440	0.6927

When K is used as a trainable parameter in the neural network, the M-IOU score was the lowest. The differences in M-IOU is due to the complexity of the data sets. The highest average mean intersection score in Table 1 leads us to selection K value of 0.3. In the next section, we will compare the Squish ($K = 0.3$) to other activation functions.

4.3 Compare Squish vs. Alternative Activation Functions

Next, we compare Squish function for tuning value $K = 0.3$ to popular activation function for five image segmentation data sets. In Table 2, we compare the M-IOU score of the Squish function to the Relu, Leaky-Relu, Elu, Selu, and Swish activation functions for each data set.

Table 2: Compare Squish ($K = 0.3$) M-IOU to alternative activation function for five image segmentation data sets.

	Pascal People	Brain MRI	Water Bodies	Oxford Flowers	Oxford Pets	Average M-IOU	Time (Sec)
Relu	0.4337	0.6955	0.7359	0.8579	0.7526	0.6951	**2207**
Leaky-Relu	0.4574	0.7066	0.7407	0.8565	0.7677	0.7058	2621
Elu	0.4375	0.7010	0.7413	0.8639	0.7423	0.6972	2346
Selu	0.4574	**0.7076**	0.7480	0.8607	0.7396	0.7027	2390
Swish	0.4535	0.7051	0.7490	0.8640	0.7590	0.7061	2579
Squish (K=0.3)	**0.4682**	0.6948	**0.7491**	**0.8648**	**0.7867**	**0.7127**	4005

We observe that the Squish ($K = 0.3$) function out preform all functions for each data set except Brain MRI. The average M-IOU score for Squish function outpreformed the Relu, Leaky-Relu, Elu, Selu, and Swish by 0.0176, 0.0069, 0.0155, 0.01, and 0.0066, respectively.

It is also important to note, the average result of most Squish functions in Table 1 out preformed the Relu and Elu activation functions. Furthermore, after Squish function, the Swish was best preforming alternative function. However, the proposed function can be limited by the computational complexity. The training time is nearly twice as long for some activation function compared Squish function.

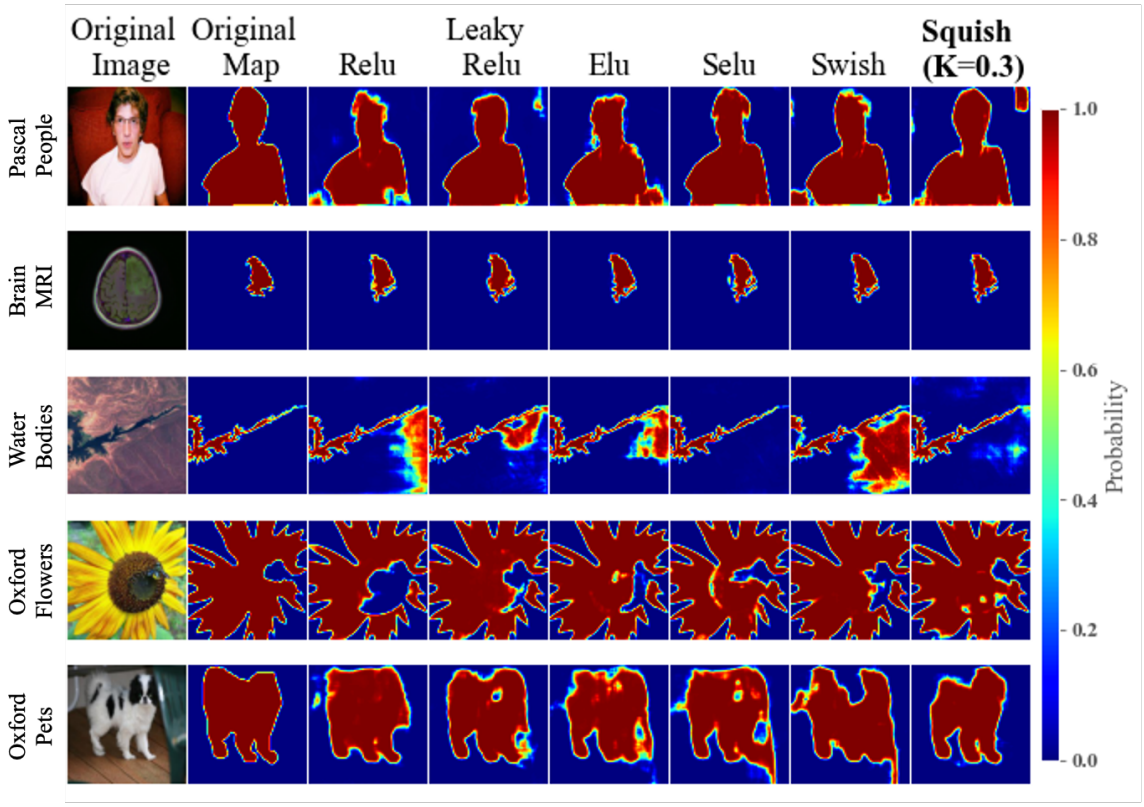

Figure 7: Compare U-Net segmentation maps of Squish function to alternative activation function for five data sets.

Predicted (output) segmentation maps are visualized in Figure 7. These segmentation maps are sampled from the predictions from each dataset, using the U-Net model with both the commonly used activation functions and the proposed Squish function. It is important to note the stronger structural shape of human head, dog body, and flower shape in pascal people, oxford pets, and oxford flowers data sets using the Squish function. In addition, the values at the center of the oxford flower image do not explode when using the Squish function, as much compared to other activation functions. Furthermore, In water body the squish function produces a significantly less noisy segmentation map, aside from Selu,

and is capable of detecting complex pattern of the river bank. Finally, for the Brain MRI Images the Squish function can also highlight strong structural alignment while detecting separations in segmentation map.

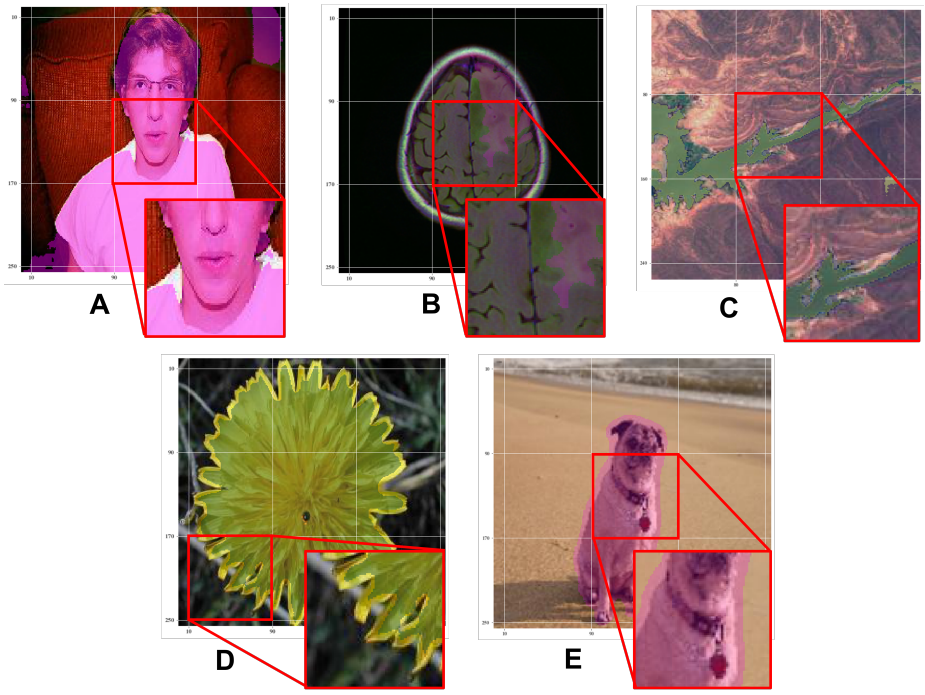

Figure 8: Local view of pixel-wise U-Net segmentation using novel Squish function.(A) Pascal People. (B) Brain MRI. (C) Water Body.(D) Oxford Flowers. (E) Oxford Pets.

In Figure 8, an enhanced view of the structure and edge detection capabilities of Squish is provided. The pixel to pixel segmentation maps show strong edge detection overlap between real image and segmentation map. In figure 8A and 8C, we can see clear overlaps of segmentation map up the edge over real image. The edge of human body, and the curves and edges within the river, are effectively captured utilizing the Squish function. Figure 8D and 8E, showcase how the predicted segmentation map matches the structure of image well, but due to the how the training segmentation map was designed, we do not see a perfect overlap of the object and the predicted segmentation map.

Lastly, Table 3 shows the some structural and noise related metrics to observe how often the functions are miss-classifying pixels. The PA highlights that the number of correctly predicted pixels are higher when using the Squish function. In addition, the F1 score shows the Squish function is preforming better for unbalanced data sets. Images with small

segmentation maps are also more likely to be predicted correctly with Squish function. Lastly, the PSNR metric shows that the Squish function outputs less noisy segmentation maps than other functions.

Table 3: Compare PA, PSNR, and F1 for Squish and alternative activation functions.

	Average PA	Average PSNR	Average F1
Relu	0.9148	36.5417	0.8413
Leaky-Relu	0.9210	37.5508	0.8493
Elu	0.9183	36.6449	0.8420
Selu	0.9186	37.5767	0.8465
Swish	0.9158	37.8472	0.8467
Squish (K=0.3)	**0.9231**	**38.5395**	**0.8528**

5 Discussion

The proposed Squish activation function has many properties that are desirable to enhance neural network learning capabilities. The theoretical properties of the proposed activation function are threefold. The function has a 1) smooth learning landscape for improved convergence with random weight initialization, 2) self-regularizing to reduce overfitting that can occur with a larger variance, and 3) controlled gradients for improved low-level feature learning. These features are achieved by thresholding and using parameter K to combine properties found in the Swish and SoftSign function. In Section 3, these properties were validated through the use of synthetically generated Gaussian kernel probability distributions and loss landscape estimations. Both of these evaluations highlight the low-level feature learning advantages and smooth loss landscape, as compared to alternative activation functions.

The proposed activation function showed improved image segmentation performance in Section 4.3, based on M-IOU, PA, PSNR, and F1 scores. However, the disadvantage of this enhanced performance is an increase in computational time. The higher computational complexity is likely due to complex gradient of the function and more mathematical operations. The self-regularizing features of Swish, and controlled gradient of the combination of Swish and SoftSign, improve the model's ability to generalize features within the images. When evaluating the Squish function as bounded vs. unbounded, it is important to note

that when the Squish is bounded in upper region, the values will self-normalize. This can lead to a smaller gradient, resulting in less low-level feature learning. This is the likely cause of the lower metric scores for the bounded Squish function. Therefore, increasing the gradient size increase the ability for the model to learn low-level features.

Furthermore, the differences in M-IOU scores are due to the complexity of images being segmented. This is highlighted by the Pascal People data set, as many of the images are dynamic, unbalanced, or noisy, which makes it difficult for any model to achieve good results with similar training settings. In addition, limited training samples will impact the performance of models leveraging the Squish function. This is another characteristic of the Pascal People data set, where minimal training samples were available, as compared to other data sets. Another observation is that balanced data sets tend to lead to better performance, such as the Oxford pets or Oxford flowers. The Oxford pet images contain one object (animal), and are usually very balanced between animal and background. This decreases the variance of the image, improving M-IOU.

Overall, the broad impact of this activation function is that we have developed a smooth, self-regularizing function that is able to extract and learn detailed low-level features, achieve enhanced edge detection, and converge more frequently to an optimal value.

6 Conclusion

Image segmentation is a challenging DL task that involves training a model to learn complex, non-linear patterns within an image, and subsequently generate a binary map of the object contained within the image. The selection of an activation function that can improve the mapping of the non-linear features within images is very valuable. The Squish activation function, and related methodology, combines the valuable aspect of two widely recognized activation function to improve image segmentation results. We expect the Squish function can be improved to reduce computational complexity while holding up the same results. Additionally, exploring a new way to train and tune combination parameter K could improve performance. Overall, the Squish combination function has a a self-regularizing, smooth landscape, and a controlled gradient, all of which are valuable properties for improving DL neural networks. These properties help to distinguish the Squish's enhanced segmentation

and edge detection performance, over other commonly used alternatives.

References

[1] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, "A survey on deep learning: Algorithms, techniques, and applications," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, pp. 1–36, 2018.

[2] K. B. Obaid, S. Zeebaree, O. M. Ahmed *et al.*, "Deep learning models based on image classification: a review," *International Journal of Science and Business*, vol. 4, no. 11, pp. 75–81, 2020.

[3] L. Cai, J. Gao, and D. Zhao, "A review of the application of deep learning in medical image classification and segmentation," *Annals of translational medicine*, vol. 8, no. 11, 2020.

[4] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[5] X. Wu, D. Sahoo, and S. C. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020.

[6] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE access*, vol. 7, pp. 128 837–128 868, 2019.

[7] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[8] E. Goceri, "Challenges and recent solutions for image segmentation in the era of deep learning," in *2019 ninth international conference on image processing theory, tools and applications (IPTA)*. IEEE, 2019, pp. 1–6.

[9] B. R. Kiran, D. M. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *Journal of Imaging*, vol. 4, no. 2, p. 36, 2018.

[10] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "A comprehensive survey and performance analysis of activation functions in deep learning," 2021. [Online]. Available: http://arxiv.org/abs/2109.14545

[11] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, vol. 30, 2017.

[12] V. N. Chitta Ranjan, Yisha Xiang and A. Rajendran, "Processminer data challenge: Activation function in deep."

[13] Y. Hu, A. Huber, J. Anumula, and S.-C. Liu, "Overcoming the vanishing gradient problem in plain recurrent networks," 2018. [Online]. Available: http://arxiv.org/abs/1801.06105

[14] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," 2017. [Online]. Available: https://arxiv.org/abs/1710.05941

[15] P. Sibi, S. A. Jones, and P. Siddarth, "Analysis of different activation functions using back propagation neural networks," *Journal of theoretical and applied information technology*, vol. 47, no. 3, pp. 1264–1268, 2013.

[16] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized mlp architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111–122, 2011.

[17] Y. Zhou, D. Li, S. Huo, and S.-Y. Kung, "Soft-root-sign activation function," 3 2020. [Online]. Available: http://arxiv.org/abs/2003.00547

[18] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," 12 2017. [Online]. Available: http://arxiv.org/abs/1712.09913

[19] M. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*, vol. 10072, 2016, pp. 234–244.

[20] F. Van Beers, A. Lindström, E. Okafor, and M. Wiering, "Deep neural networks with intersection over union loss for binary image segmentation," in *ICPRAM*, 2019.

[21] F. PirahanSiah, S. N. H. S. Abdullah, and S. Sahran, "Adaptive image segmentation based on peak signal-to-noise ratio for a license plate recognition system," in *2010 International Conference on Computer Applications and Industrial Electronics*, 2010, pp. 468–472.

[22] F. Kromp, L. Fischer, E. Bozsaky, I. M. Ambros, W. Dörr, K. Beiske, P. F. Ambros, A. Hanbury, and S. Taschner-Mandl, "Evaluation of deep learning architectures for complex immunofluorescence nuclear image segmentation," vol. 40, no. 7, 2021, pp. 1934–1949.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results." [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[24] M. Buda, "LGG Segmentation Dataset." [Online]. Available: https://www.kaggle.com/datasets/mateuszbuda/lgg-mri-segmentation

[25] F. Escobar, "Satellite images of water bodies dataset." [Online]. Available: https://www.kaggle.com/datasets/franciscoescobar/satellite-images-of-water-bodies

[26] A. Z. Maria-Elena Nilsback, "17 category flower dataset." [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

[27] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Oxford-IIIT pets dataset." [Online]. Available: https://www.robots.ox.ac.uk/~vgg/data/pets/