# Deep Learning CS-671 Hackathon

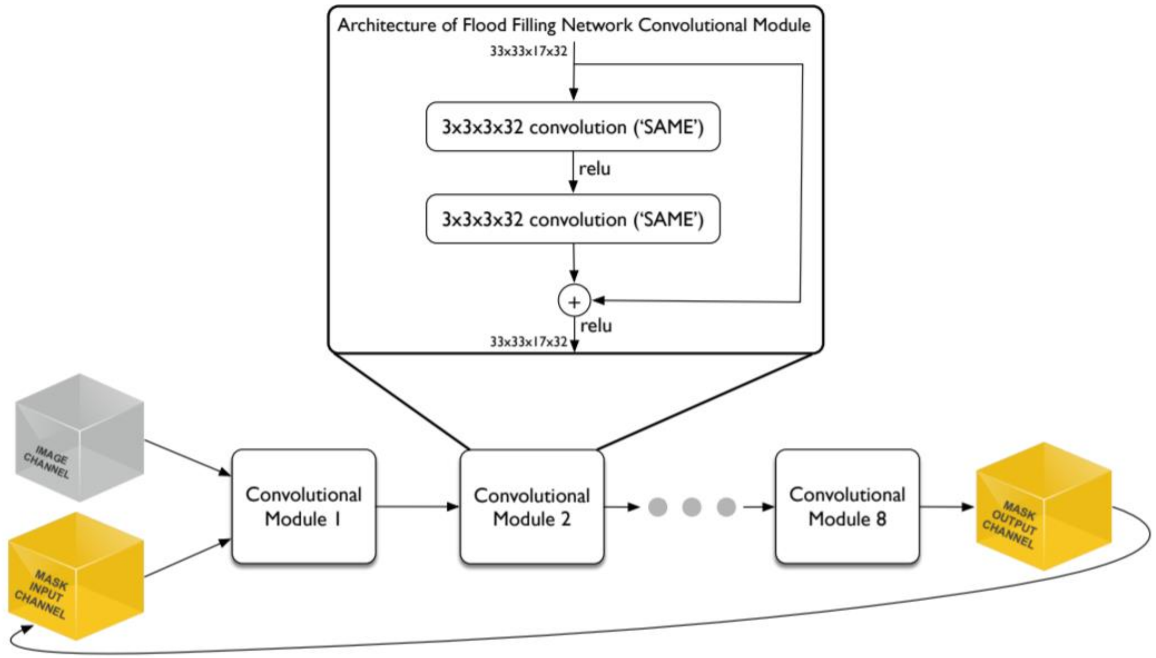
**Optimizing Image Segmentation using Flood Filling Networks**

# A Different Approach

- We look to implement the segmentation approach of "flood-filling networks" (FFNs).
- Forgoes the multiple steps of boundary detection and subsequent segmentation.
- Uses a single network, analogous to a RNN, to process raw image pixels directly into individual object masks.
- Approach has been validated on a strenuous three dimensional image segmentation task - connectomic reconstruction from volume electron microscopy data, on which flood-filling neural networks proved to be a significant improvement in terms of accuracy over other state-of-the-art methods.

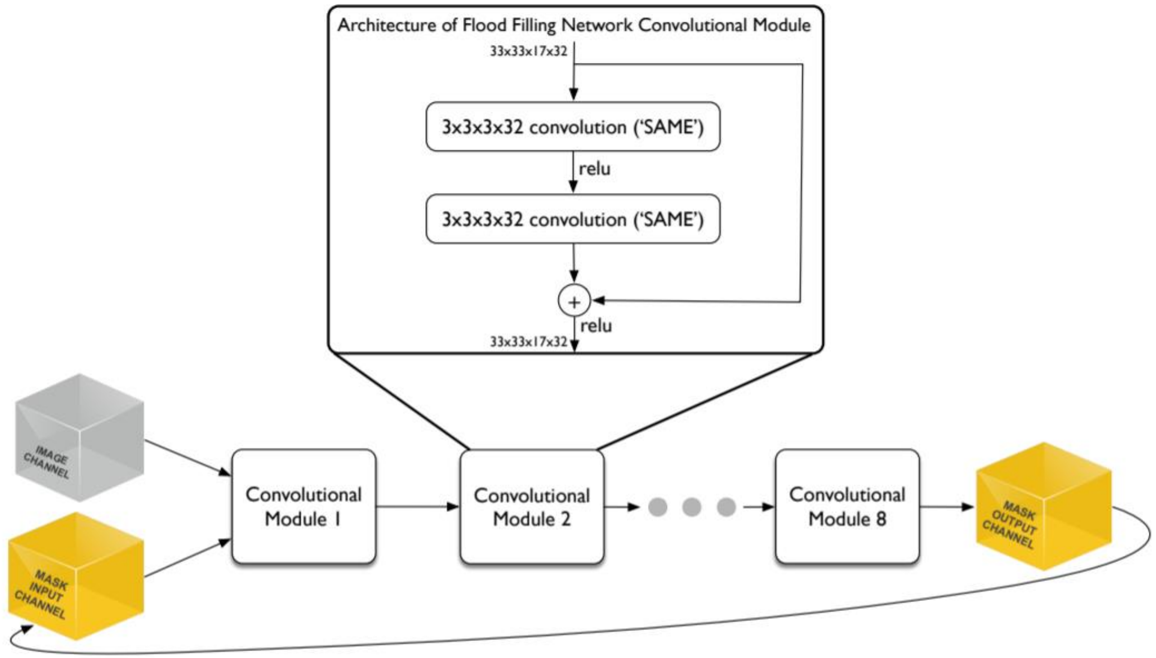
# Flood Filling Networks

- A traditional flood-filling network takes a three dimensional sub volume of data as input and produces an object mask probability map.



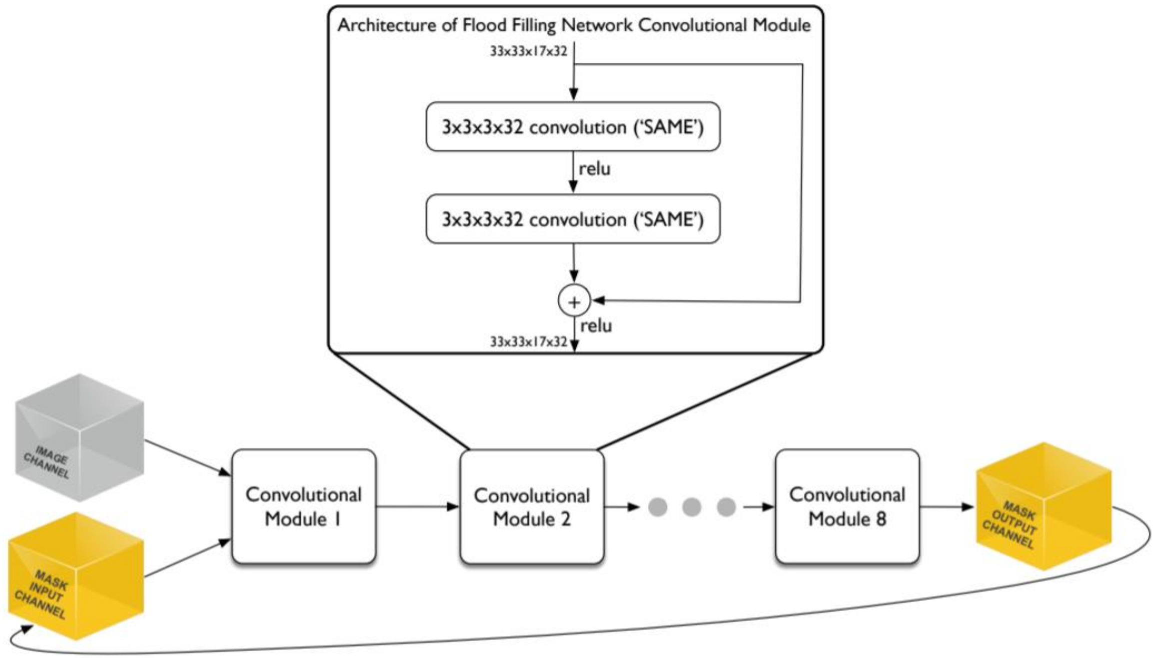
# Flood Filling Networks

- The input sub volume contains two channels: one providing the raw image intensities, and another providing the local state of the object mask in the form of a probability map.



# Flood Filling Networks

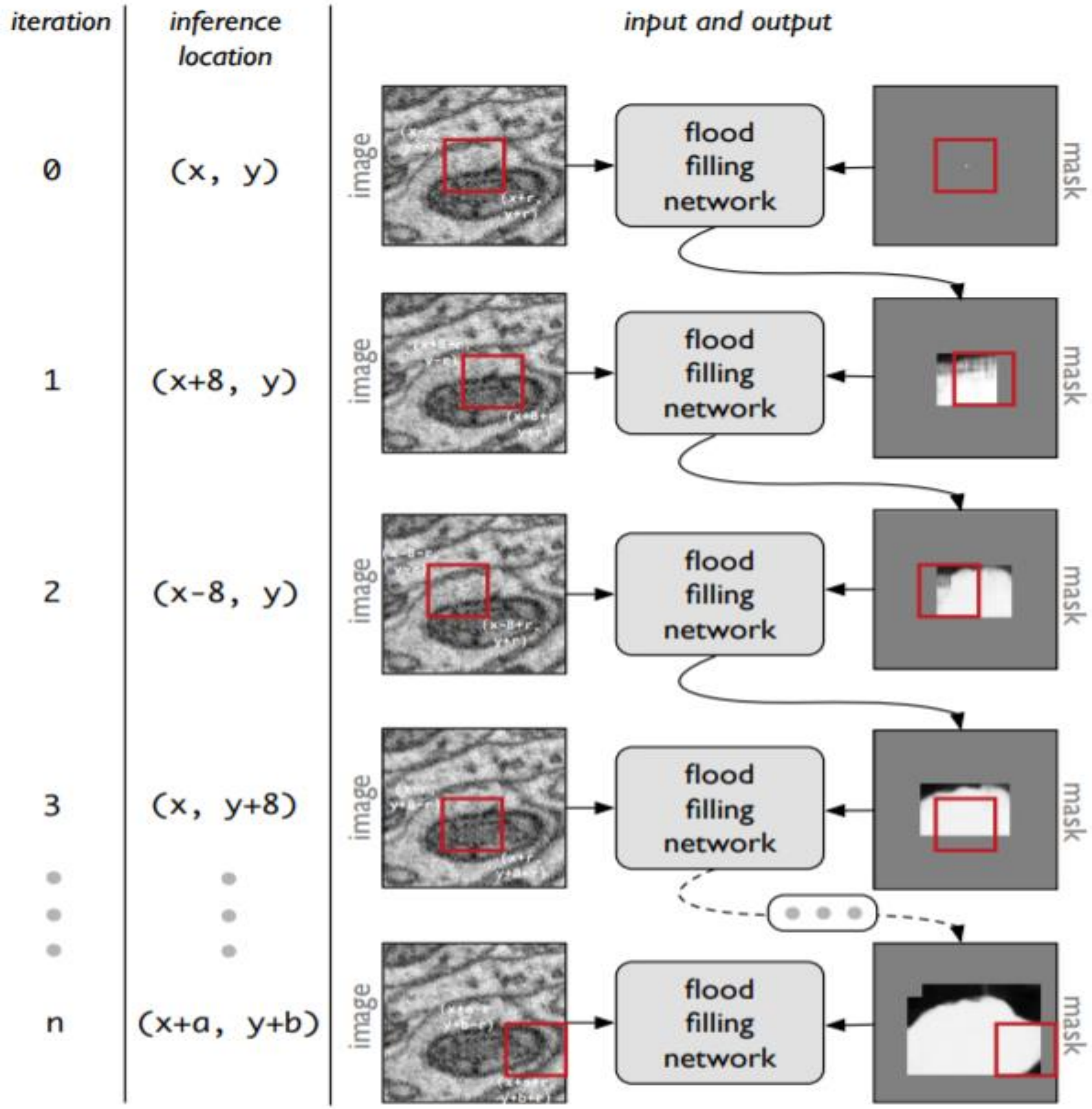
- Flood filling networks have been developed to segment objects over n number of iterations. FFN start with one pixel and keep on iterating till a final segmentation is produced.



# Flood Filling Networks

- The input mask in the second channel is incomplete in most inference iterations of the network.
- The FFN is trained to extend it within its field of view (FoV).
- At the beginning of inference, in absence of any prior information about object shape, the incomplete object mask can consist of a single active voxel.
- The network then performs single-object segmentation for the object covered by that voxel.

# Schematic of multiple-field-of-view inference of a flood-filling network.



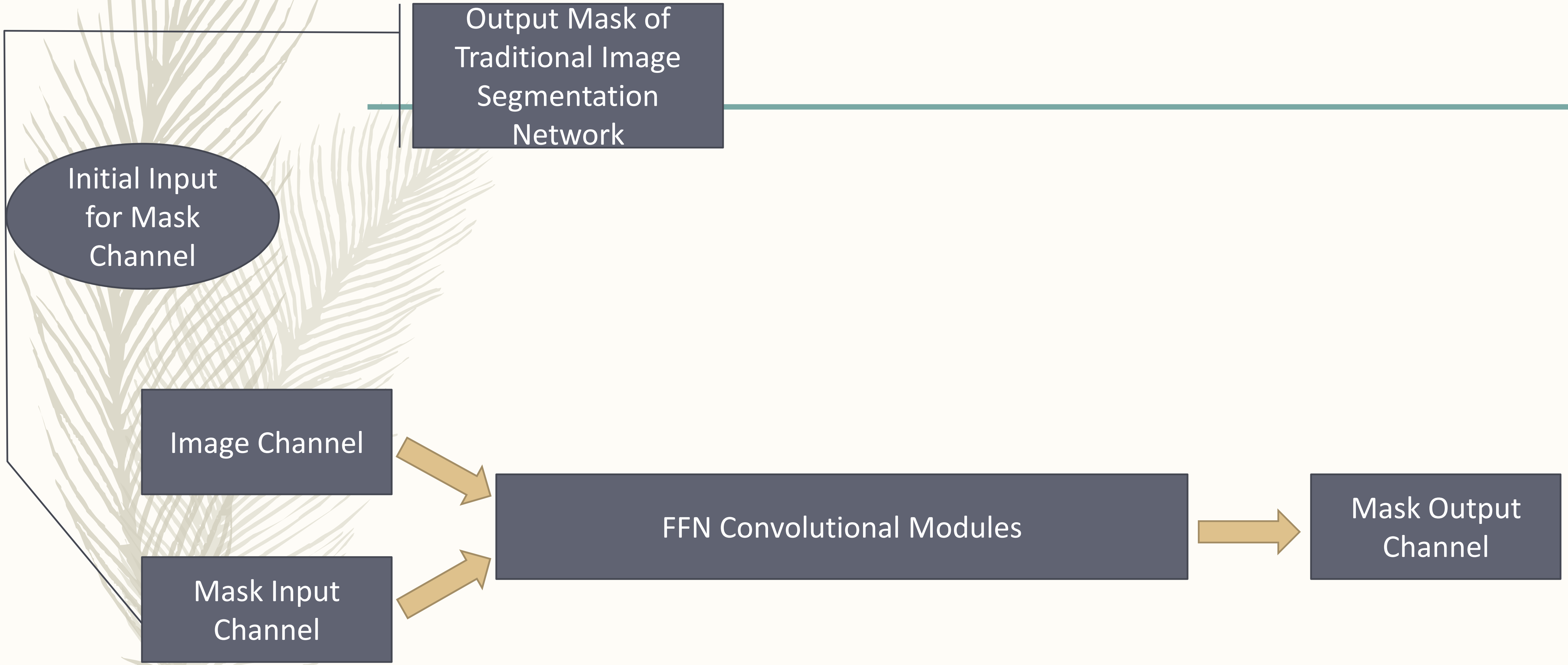
Observed Improvements of FFN technique on an electron microscopy problem.

| Segmentation | Edge accuracy [%] | Merged edges [%] | Split edges [%] | Omitted edges (adjusted) [%] | Omitted edges (raw) [%] |
|---|---|---|---|---|---|
| CNN + Watershed | 87.7 | 1.0 | 10.6 | 0.7 | 1.1 |
| CNN + Watershed + GALA | 96.3 | 1.7 | 1.4 | 0.6 | 1.1 |
| CNN + Watershed + CELIS | 93.2 | 5.4 | 0.7 | 0.7 | 1.1 |
| Flood-Filling Network | 98.5 | 0.0 | 0.7 | 0.8 | 2.4 |

# Our Architecture

- FFN start with one pixel and keep on iterating till a final segmentation is produced.
- What we propose is that using a simple traditional neural –net to first produce a mask and then this mask be fed to the FFN.
- Thus, output of the initial neural network serves as a much better starting point than a single pixel.
- It also efficiently tackles the problem of slow rate of segmentation progress in initial iterations of a flood filling network, as the FFN now has prior information about image at onset of operation of the FFN.
- Accuracy of the initial neural network would be improved upon.

# Our Architecture



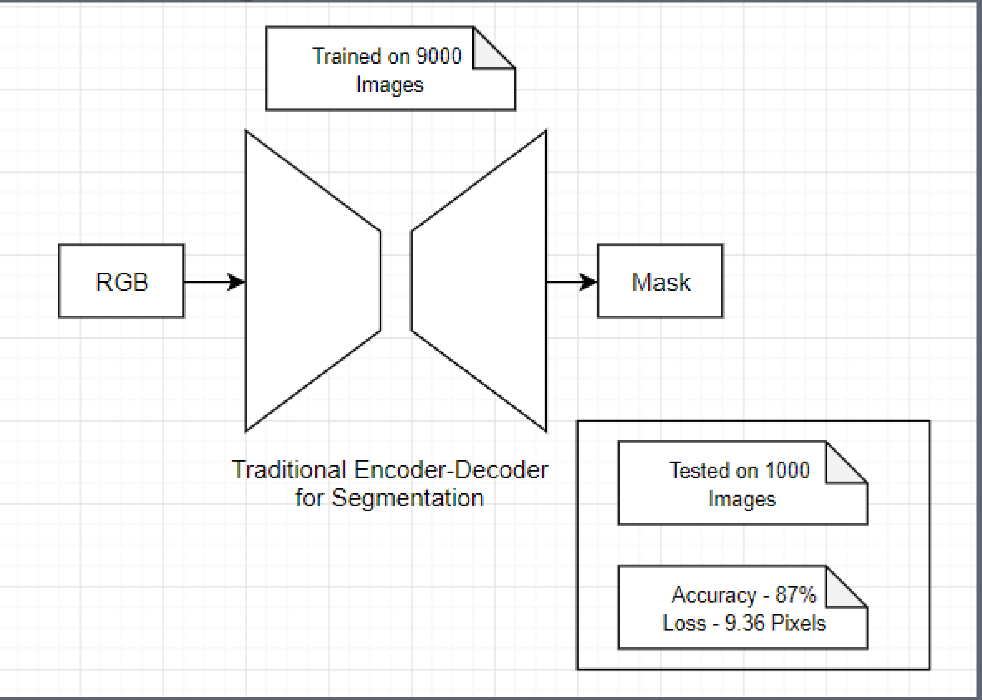
# Our Architecture

Output Mask of Traditional Image Segmentation Network

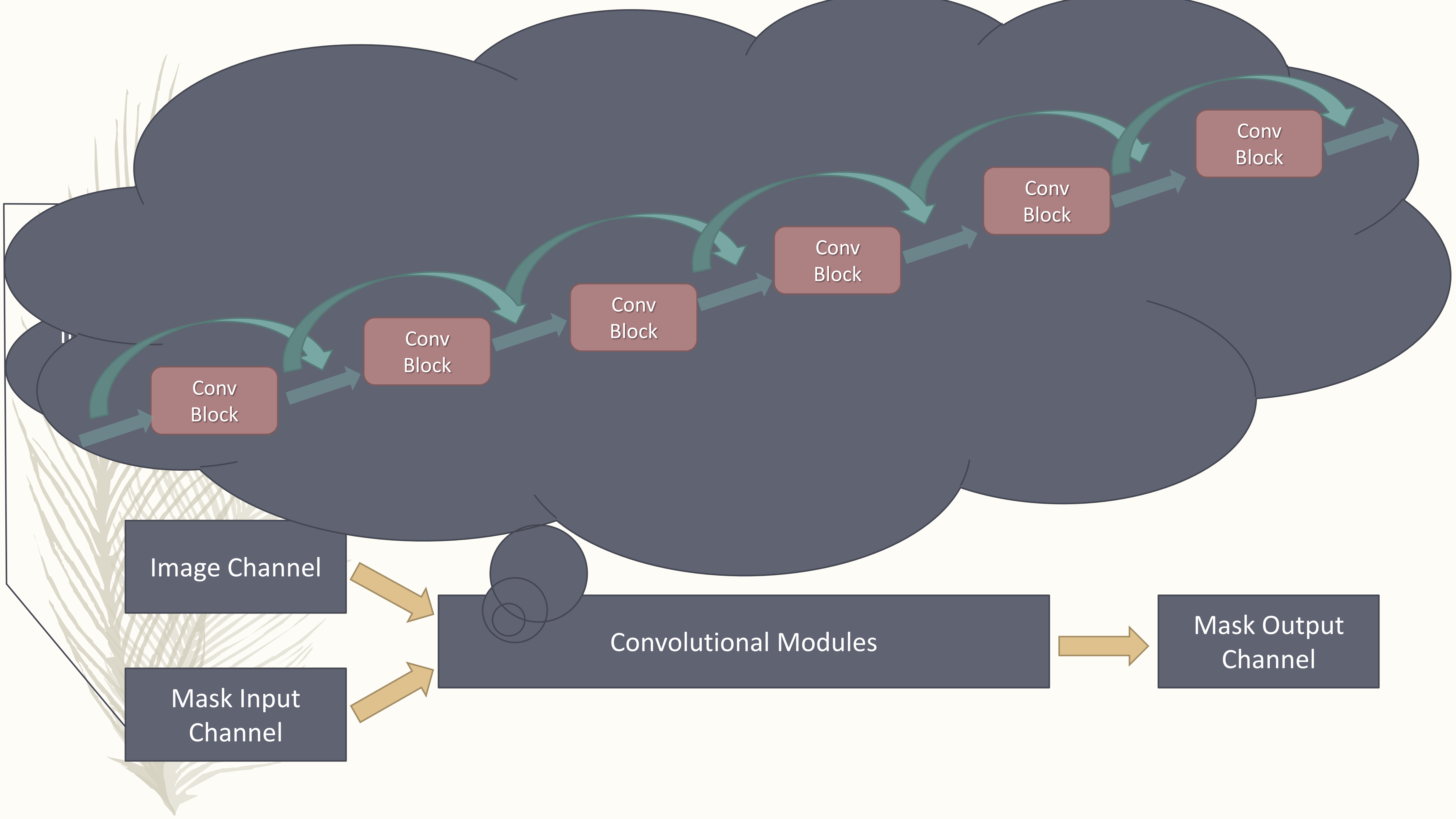
Initial Input for Mask Channel

Image Channe

Mask Input Channel

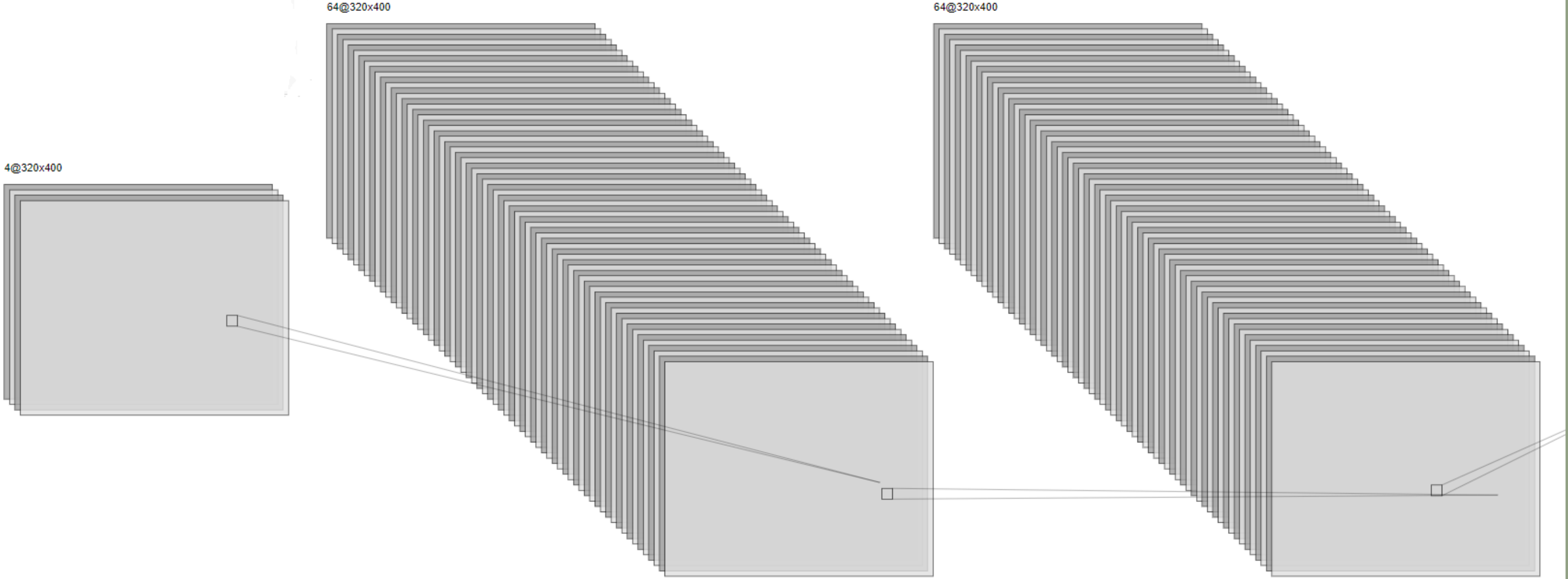



Output Channel



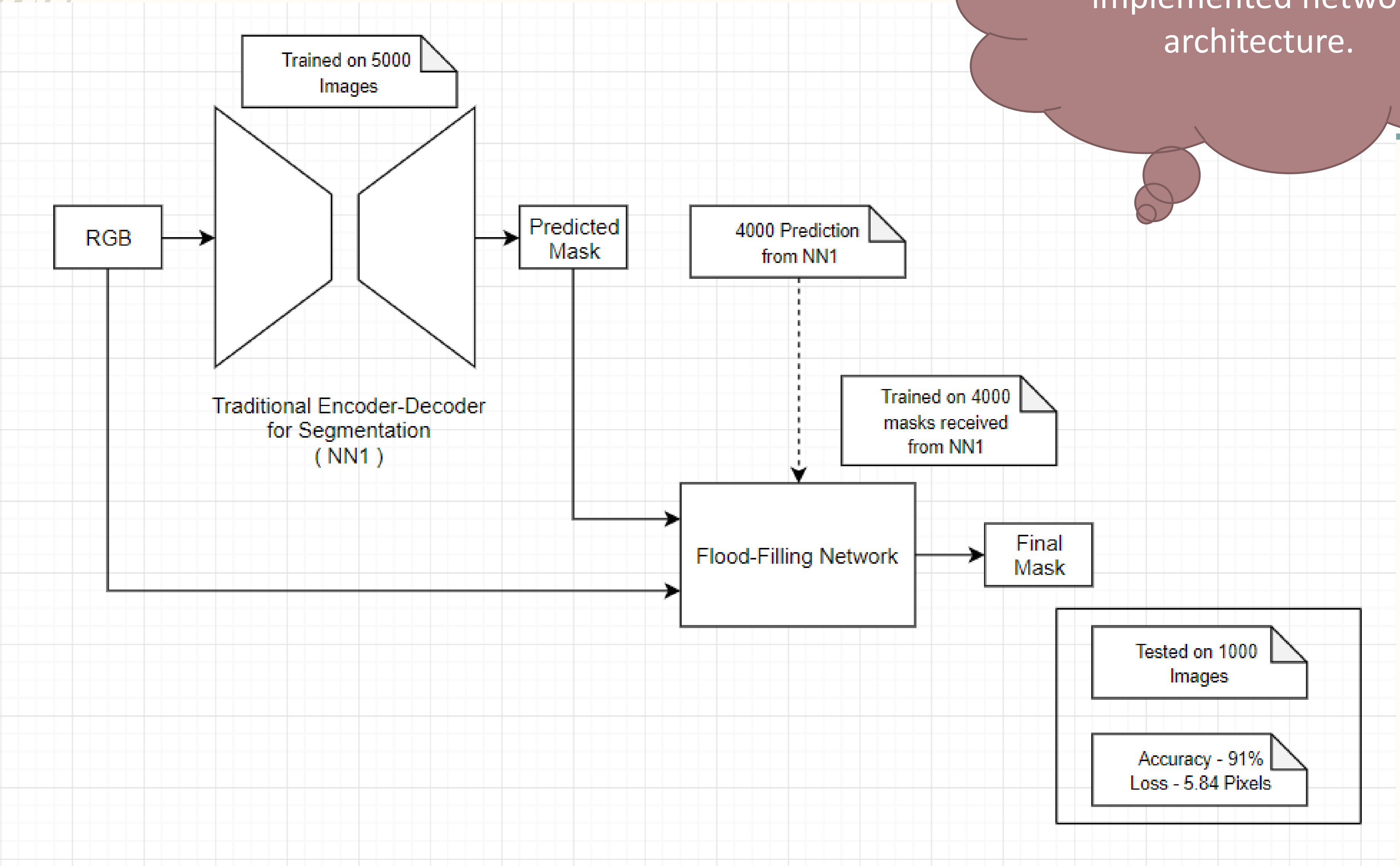
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Conv Block
Image Channel
Mask Input Channel
Convolutional Modules
Mask Output Channel

# Architecture of one block



# Our Architecture

Workflow of our implemented network architecture.



# Our Architecture

Proposed Benefits:

- The FFN doesn’t have to start with one-pixel as a mask input.
- The traditional neural net need not be a complex neural net.
- Our implementation of FFN can be used to increase accuracy of underperforming segmentation models.

# Observations

- The iris dataset was used as the training and testing data.
- We used 9000 images to train the first encoder-decoder network, and subsequently tested 1000 images.
- The second network, that is, the FFN was trained on 4000 images received from NN1, and subsequently tested on the same 1000 images.
- On observation, the first network achieved accuracy of 87% and loss of 9.36 pixels.
- On observation, our FFN implementation achieved accuracy of 91% and loss of 5.84 pixels.

# Inferences

- Training complexity of our architecture is observed to be significantly less than that of our traditional model which was trained on 9000 images.
- End to end training is not feasible in our architecture. Thus, the NN1 was trained first, and subsequently

# End of Presentation

Thank You

Hrishikesh, B16029

Karan, B16020

Sammarth, B16031