

# What makes a flower its kind.

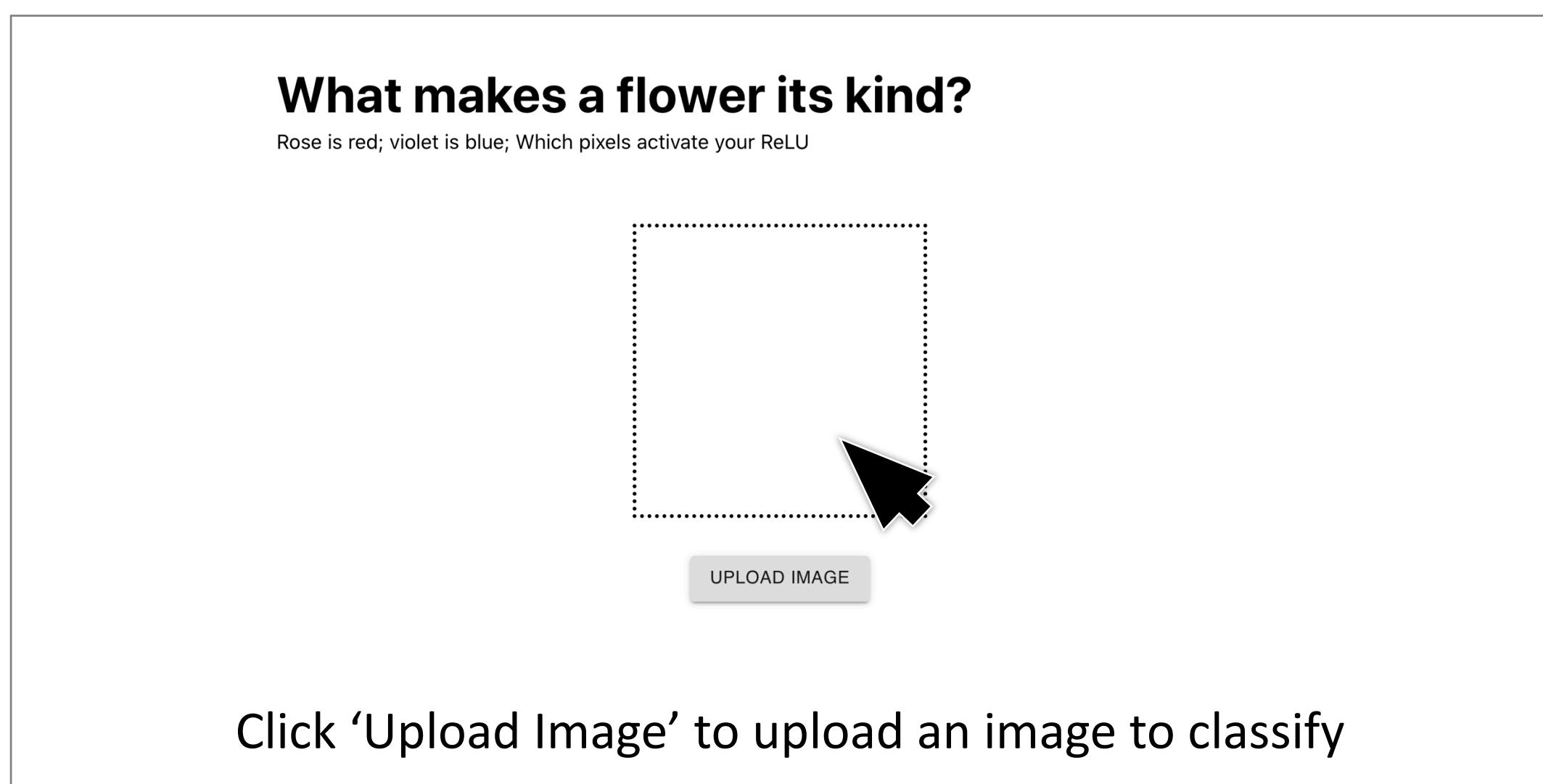
Interactive visualization tool for visualizing layout of Convolutional Neural Network

Chanwut Kittivorawong, Podshara Chanrungmaneekul



PAUL G. ALLEN SCHOOL  
OF COMPUTER SCIENCE & ENGINEERING

## Usability



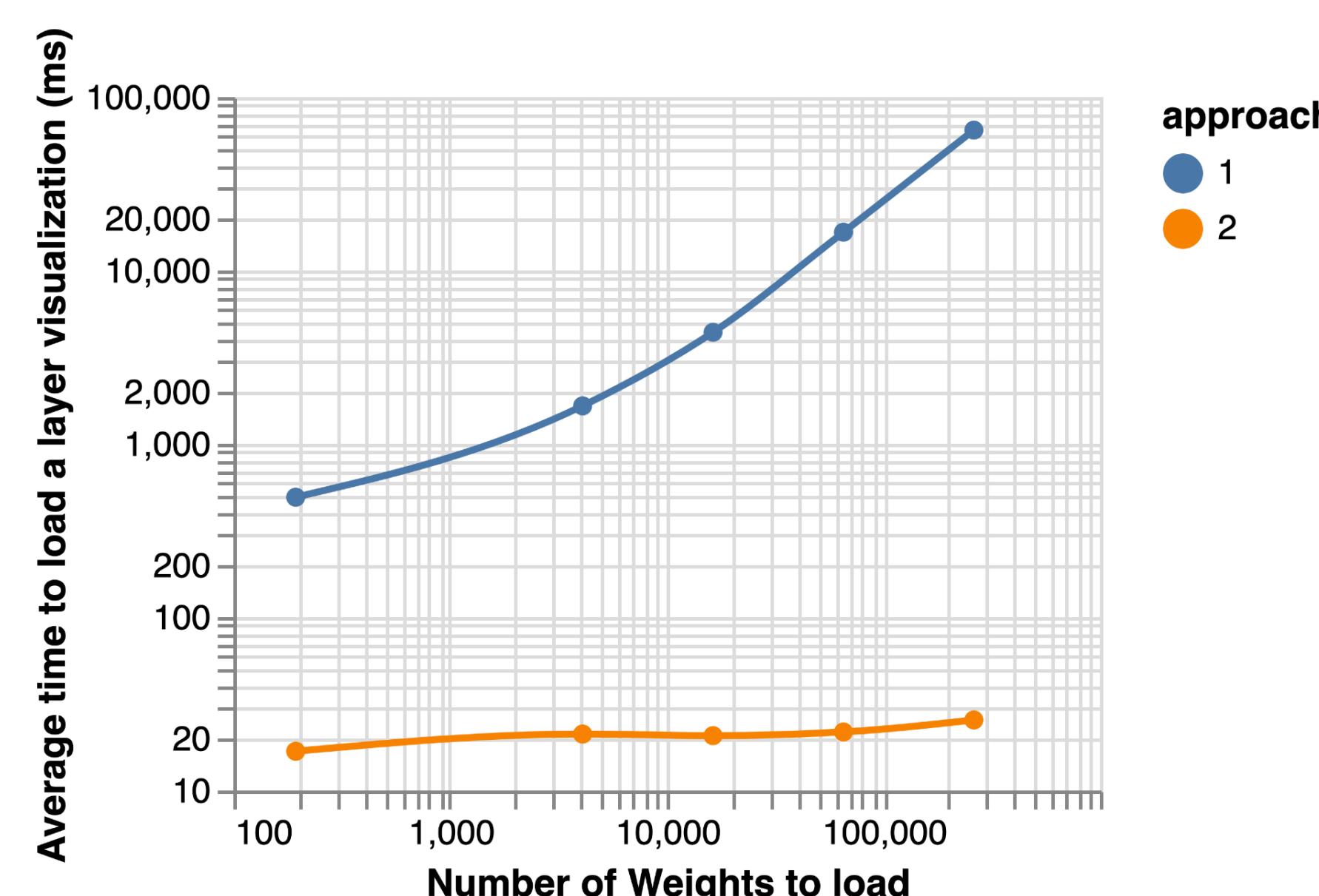
Click 'Upload Image' to upload an image to classify

## Introduction

Learning to understand the convolutional neural network is difficult for beginners who do not have much experience with machine learning and computer vision and it is hard what is going on between each layer that makes the network come up with the output. For this reason, we created a tool that can visualize a sequence of convolutional layers in a neural network written using PyTorch, specifically their inputs, outputs, and weights. This tool helps beginners to start learning neural network by understanding how a neural network "think" of the input image through the visualization of each layer, as well as a useful debugger tool to look inside a network when something goes wrong.

## Performance

We have tried two approaches to calculate and display weights of convolutional layers, to give the best performance in the visualization. For the first approach, every time users click to visualize each layer, the application loads all the weights in the layer and convert them into images. For the second approach, the application loads each weight lazily; that is, it only loads and convert weight into an image only when requested. Here is the result from performance testing



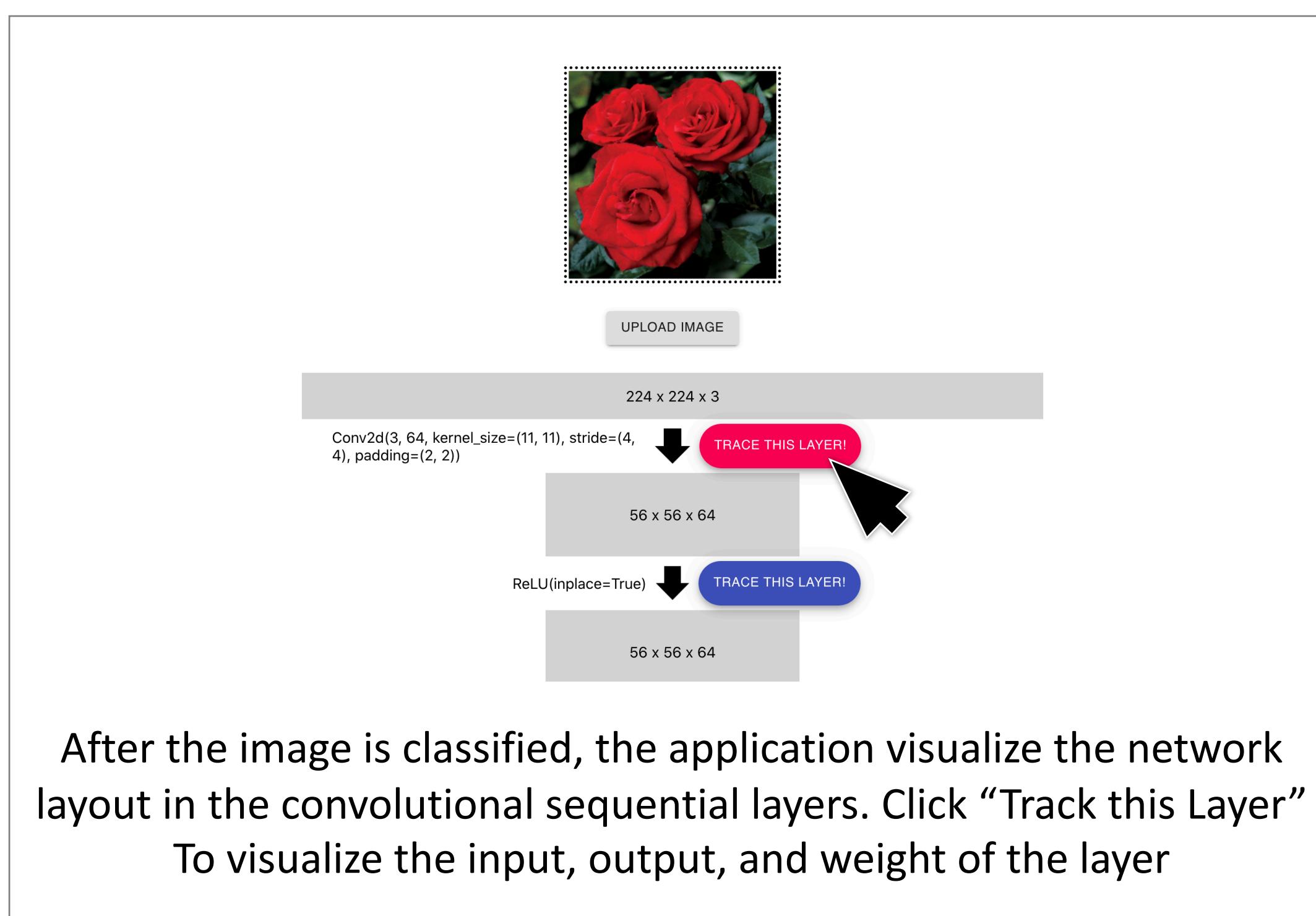
We choose to go with the second approach. Even though the loading time for a single weight for the second approach is more than that of the first approach the difference is almost unnoticeable in real-world usage (see table above), but it reduces a big portion of initial loading time when loading the visualization for each layer.

Table comparing the performance of loading visualization and loading single weight of both approaches.

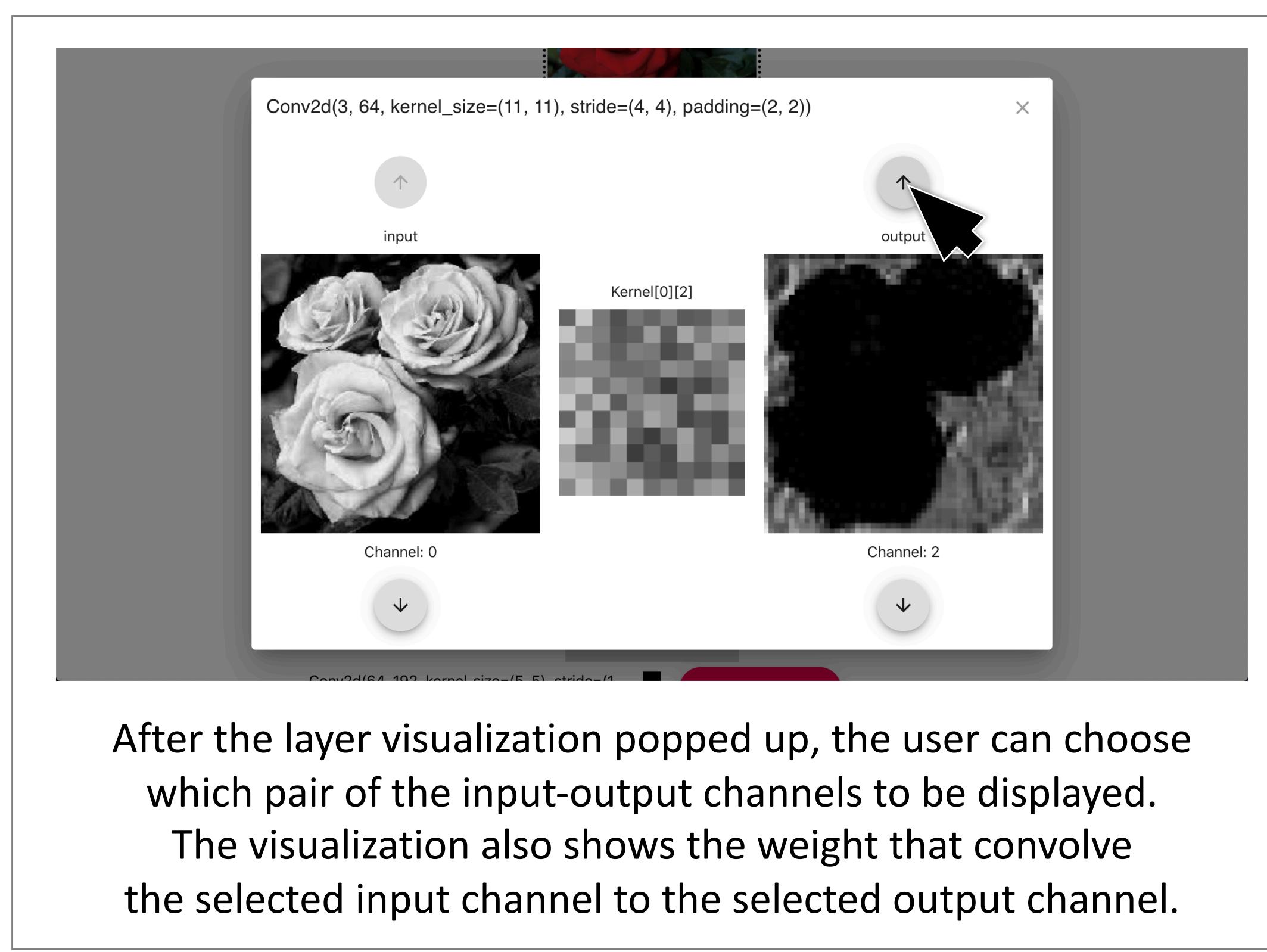
Number of weights	Time to load a visualization for one layer (ms)		Time to load a single weight (ms)	
	First approach	Second approach	First approach	Second approach
3 x 64	492	464	0 to 2	16.84
64 x 64	1656	512	0 to 2	21.18
128 x 128	4400	440	0 to 2	20.71
256 x 256	16624	494	0 to 2	21.77
512 x 512	64551	706	0 to 2	25.52

## Modularity

One of the main reason we build this tool is to have a generalizable visualizer that works across sequential CNNs. We have created a VisualizableSequential module that can be used to replace the original network's own Sequential of convolutional layers. Then, it extract the input, output, and weights to visualize them when plugging the network into this application.



After the image is classified, the application visualize the network layout in the convolutional sequential layers. Click "Track this Layer" To visualize the input, output, and weight of the layer



After the layer visualization popped up, the user can choose which pair of the input-output channels to be displayed. The visualization also shows the weight that convolve the selected input channel to the selected output channel.

