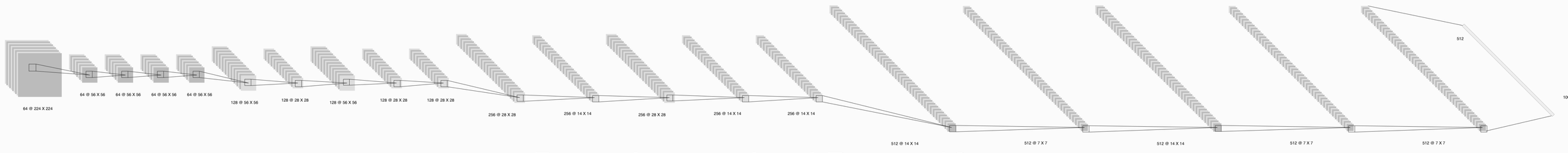
Automating the creation of Neural Network Architecture Diagrams from Code

Eunice Jun *emjun@uw.edu*
Matthew Conlen *mconlen@uw.edu*

DESIGN USE BUILD
PAUL G. ALLEN SCHOOL OF COMPUTER SCIENCE & ENGINEERING



Pytorch pretrained Resnet18

Motivation

(1) Architecture diagrams support efficient communication of complex Deep Learning models. However, **current visualizations** are **inconsistent** and **difficult to compare** across sources due to a lack of standards and tools.

(2) Architecture diagrams aid in the creation, debugging, and understanding of models. However, **current tools** for visualization are **detached from the development process** and introduce significant time and effort into the development process.

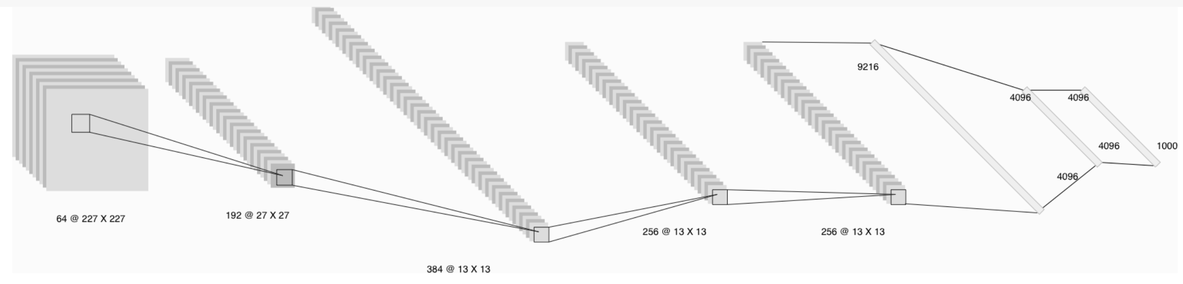
Design

NNView determines the structure of models through inspection at runtime when possible, and running a single forward-pass through the network otherwise. This allows us to generate architecture diagrams for arbitrary models, produced with **Pytorch** and **Keras***.*

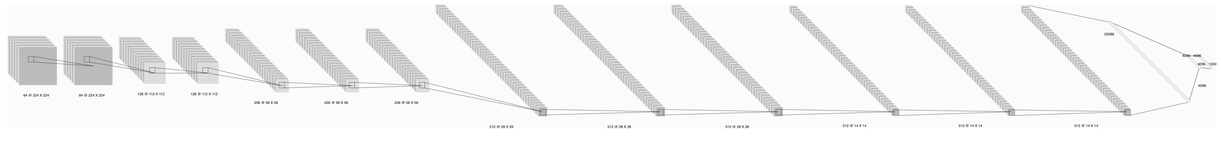
NNView generates **Scalable Vector Graphic** (SVG) output which is displayed directly in **computational notebooks** (Jupyter, Colab) and can be opened in programs like Adobe Illustrator for refinement before publication.

Examples

```
import NNView as nnv
net = models.alexnet()
net_input = torch.randn(1, 3, 227, 227)
ptm = nnv.models.PyTorchModel(net, net_input)
ptm
```



Pytorch pretrained AlexNet



Pytorch pretrained VGG16

Conclusions & Future Work

NNView can support machine learning education. We hope to evaluate the usefulness of in situ visualizations of neural networks with novices, and are interested to see how our work impacts the development of mental models and intuitions around neural networks.

We are interested in exploring how to improve existing model diagrams with better visual encodings, annotation, and interactivity. We hope to incorporate support for **MXNet** models in *NNView.*