MLPs resources:

- Check out the **first research paper** to propose dropout as a technique for overfitting.
- Here's the Keras **documentation** for the Flatten layer.
- If you'd like more information on activation functions, check out this **website**.

Applications of CNNs:

- Read about the **WaveNet** model.
  - Why train an A.I. to talk, when you can train it to sing ;)? In April 2017, researchers used a variant of the WaveNet model to generate songs. The original paper and demo can be found **here**.
- Learn about CNNs **for text classification**.
  - You might like to sign up for the author's **Deep Learning Newsletter**!
- Read about Facebook's **novel CNN approach** for language translation that achieves state-of-the-art accuracy at nine times the speed of RNN models.
- Play **Atari games** with a CNN and reinforcement learning. You can **download** the code that comes with this paper.
  - If you would like to play around with some beginner code (for deep reinforcement learning), you're encouraged to check out Andrej Karpathy's **post**.
- Play **pictionary** with a CNN!
  - Also check out all of the other cool implementations on the **A.I. Experiments** website. Be sure not to miss **AutoDraw**!
- Read more about **AlphaGo**.
  - Check out **this article**, which asks the question: *If mastering Go "requires human intuition," what is it like to have a piece of one's humanity challenged?*
- Check out these *really cool* videos with drones that are powered by CNNs.
  - Here's an interview with a startup - **Intelligent Flying Machines (IFM)**.
  - Outdoor autonomous navigation is typically accomplished through the use of the **global positioning system (GPS)**, but here's a demo with a CNN-powered **autonomous drone**.
- If you're excited about using CNNs in self-driving cars, you're encouraged to check out:
  - our **Self-Driving Car Engineer Nanodegree**, where we classify signs in the **German Traffic Sign** dataset in **this project**.

    - our **Machine Learning Engineer Nanodegree**, where we classify house numbers from the **Street View House Numbers** dataset in **this project**.
    - this **series of blog posts** that details how to train a CNN in Python to produce a self-driving A.I. to play Grand Theft Auto V.
- Check out some additional applications not mentioned in the video.
    - Some of the world's most famous paintings have been **turned into 3D** for the visually impaired. Although the article does not mention *how* this was done, we note that it is possible to use a CNN to **predict depth** from a single image.
    - Check out **this research** that uses CNNs to localize breast cancer.
    - CNNs are used to **save endangered species**!
    - An app called **FaceApp** uses a CNN to make you smile in a picture or change genders.

Groundbreaking CNN Architectures:

- Check out the **AlexNet** paper!
- Read more about **VGGNet** here.
- The **ResNet** paper can be found here.
- Here's the Keras **documentation** for accessing some famous CNN architectures.
- Read this **detailed treatment** of the vanishing gradients problem.
- Here's a GitHub **repository** containing benchmarks for different CNN architectures.
- Visit the **ImageNet Large Scale Visual Recognition Competition (ILSVRC)** website.

Interpreting CNNs and convolutional layers:

- Here's a **section** from the Stanford's CS231n course on visualizing what CNNs learn.
- Check out this **demonstration** of a cool **OpenFrameworks** app that visualizes CNNs in real-time, from user-supplied video!
- Here's a **demonstration** of another visualization tool for CNNs. If you'd like to learn more about how these visualizations are made, check out this **video**.

- Read this **Keras blog post** on visualizing how CNNs see the world. In this post, you can find an accessible introduction to Deep Dreams, along with code for writing your own deep dreams in Keras. When you've read that:
  - Also check out this **music video** that makes use of Deep Dreams (look at 3:15-3:40)!
  - Create your own Deep Dreams (without writing any code!) using this **website**.
- If you'd like to read more about interpretability of CNNs:
  - Here's an **article** that details some dangers from using deep learning models (that are not yet interpretable) in real-world applications.
  - There's a lot of active research in this area. **These authors** recently made a step in the right direction.

Transfer Learning

- Here's the **first research paper** to propose GAP layers for object localization.
- Check out this **repository** that uses a CNN for object localization.
- Watch this **video demonstration** of object localization with a CNN.
- Check out this **repository** that uses visualization techniques to better understand bottleneck features.
- Check out this research paper that systematically analyzes the transferability of features learned in pre-trained CNNs.
- Read the Nature publication detailing Sebastian Thrun's cancer-detecting CNN!

	SIMILAR TO IMAGENET	**DIFFERENT TO IMAGENET**
SMALL	• Slice end of the neural network • Add a new fully connected layer • Randomize weights of new fully connected layer • Freeze weights of inner layers with weights from the pre-trained network • Train network to update the weights of the new fully connected layer	• Slice most of the pre-trained layers • Add a new fully connected layer • randomize weights of new fully connected layer • Freeze weights from the pre-trained network • Train network to update the weights of the new fully connected layer
LARGE	• Slice end of the neural network • Add a new fully connected layer • Randomize weights of new fully connected layer • Initialize weights of inner layers with weights using pre-trained weights • Train entire network	• Slice end of the neural network • Add a new fully connected layer • Randomize weights of all layers • Train entire network OR • Follow same approach as large & similar