**Objetivo da implementação**

Construir um classificador de imagem baseado na rede convolucional usando o Tensorflow junto com o OpenCV. Esta rede é uma rede muito pequena, assim o nosso objetivo é demonstrar como construir uma rede neural convolucional do mundo real usando o Tensorflow. Começamos cobrir os fundamentos da rede neural convolucional em seguida é detalhada a implementação finalizando com os resultados.

**Base de dados:**

Dígitos manuscritos:

Considerada uma boa base de dados para experimentar técnicas de aprendizado e métodos de reconhecimento de padrões em imagens do mundo real, enquanto gastam esforços mínimos em pré-processamento e formatação. Existem várias bases com essa característica como MNIST. Porém para este trabalho usamos o conjunto contidos no <https://github.com/kensanata/numbers> nele possui mais de 10000 dígitos separados em diretórios de acordo com certas características que identificam as pessoas como sexo faixa de idade e cidadania. No caso foi selecionado dois destes diretórios: 0024\_IT3M (60 exemplos por dígitos) é da itália pessoas com sexo masculino e com uma faixa entre 20 e 30 anos e usamos alguns dígitos selecionados por nós (Todas as imagens possuem o mesma dimensão 100x100 pixel).

Hamtaro ou Pikachu?

Aplicamos o classificador em um outro conjunto de dados criados por nós… que compõem 320 imagens de dois personagens de desenhos animados: Pikachu e Hamtaro 160 imagens de cada (Todas as imagens possuem o mesma dimensão 100x100 pixel) …

Usamos 80% dos dados serão usados para treinamento e 80% para treinamento. Baseamos no MNIST que possue 60000 para treinamento e 10000 para teste.

**Noções básicas de rede neural convolucional (CNN):**

**(Tem muita coisa aqui** [**https://www.tensorflow.org/tutorials/layers**](https://www.tensorflow.org/tutorials/layers)**)**

Tipos de camadas que compõem uma CNN:

Camada Convolucional

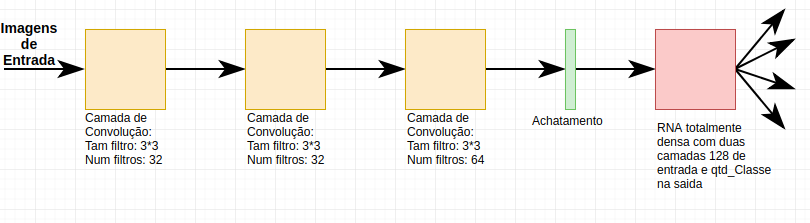
Camada de Pooling

Camada Totalmente Conectada (RNA)

**Funcionamento do Projeto**  
 Todos os dados de treinamento são armazenados em uma pasta ./data/{dataset\_name}/{classes} Quando um processo de treinamento é iniciado, ele armazena seu modelo na pasta ./model/ Uma vez que o modelo é construído, ele é usado para uma previsão adicional. As imagens para previsão são diferentes das imagens de treino. Eles são armazenados em uma subpasta ./data/{dataset\_name}/predict/. A primeira letra de um nome de arquivo é realmente o nome da classe usada para verificar os resultados. Os resultados são printados na tela armazenados em uma página HTML com imagem avaliada junto com o resuktado gerados pela rede e a saída esperada.

**Construção da rede:**

Durante o treino, as imagens de ambas as classes são fornecidas a uma camada convolucional que é seguida por mais 2 camadas convolucionais. Após as camadas convolucionais, “achatamos” a saída e adicionamos duas camadas totalmente conectadas no final. A segunda camada totalmente conectada é a camada de saída.



Usou-se a função de ativação softmax para saída e a RELU para as demais camadas.

Essas configurações foram baseadas no tutorial do tensorflow: <https://www.tensorflow.org/tutorials/layers>