Convolutional Neural Networks

Completed by Mohamad Ahmad

Related Theory

Convolutional neural networks (CNNs) are a constrained version of standard fully-connected feed-forward (FC) networks. Rather than keeping all the edges, **we modify the architecture and share some weights**. One may be inclined to think that setting many weights to 0 may increase the risk obtained by the standard FC network, however this modification of architecture to yield shared weights and local connections is better able to take advantage of the **full 2D structure** of an image (Alzubaidi et al., 2021).

Given that we are still in the domain of neural networks, it is no surprise that the CNN takes **great inspiration from neuroscience**. In fact, much of how a cat's visual cortex functions is simulated by a CNN (Hubel & Wiesel, 1962). Another great connection between the human brain and CNNs is found in developmental neuroscience. In particular, human babies first show great interest for **high contrast regions** such as edges. They scan over these edges repeatedly so as to keep brain cells highly activated. Over time, the baby will analyze more complex visual stimuli which are composed of several of these high contrast regions (Younger et al., 2012).

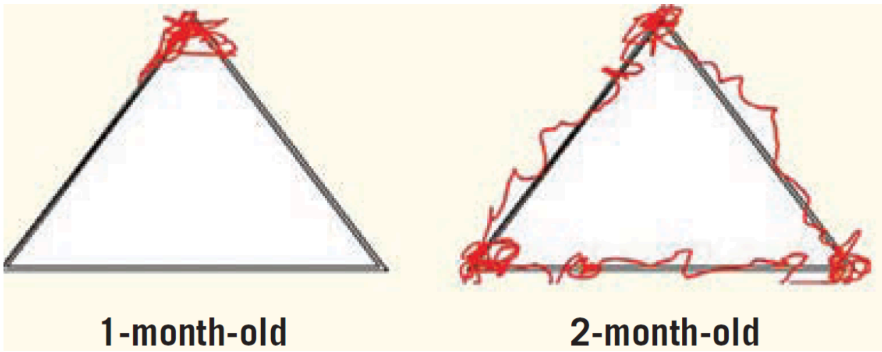


Figure 1: Visual scanning patterns of where a newborn looks at 1 month (left) and 2 months old (right). Adapted from “Visual scanning of geometric figures by the human newborn” (Salapatek, 1968).

As we will see, this progression of human vision is remarkably similar to how the CNN looks for higher-complexity features in deeper and deeper layers of the network. To derive the CNN, we posit that **analytic vision considers specific regions of an image**. These are called “patches”. We also note that depending on the image, **these patches could appear in different regions**. Hence, we should look for specific features at different possible regions (e.g. Albert Einstein's hair can be identified regardless of translations of images of Albert Einstein). A helpful mathematical tool is hence the convolution operation, consisting of sliding a matrix (kernel) over another (the image).

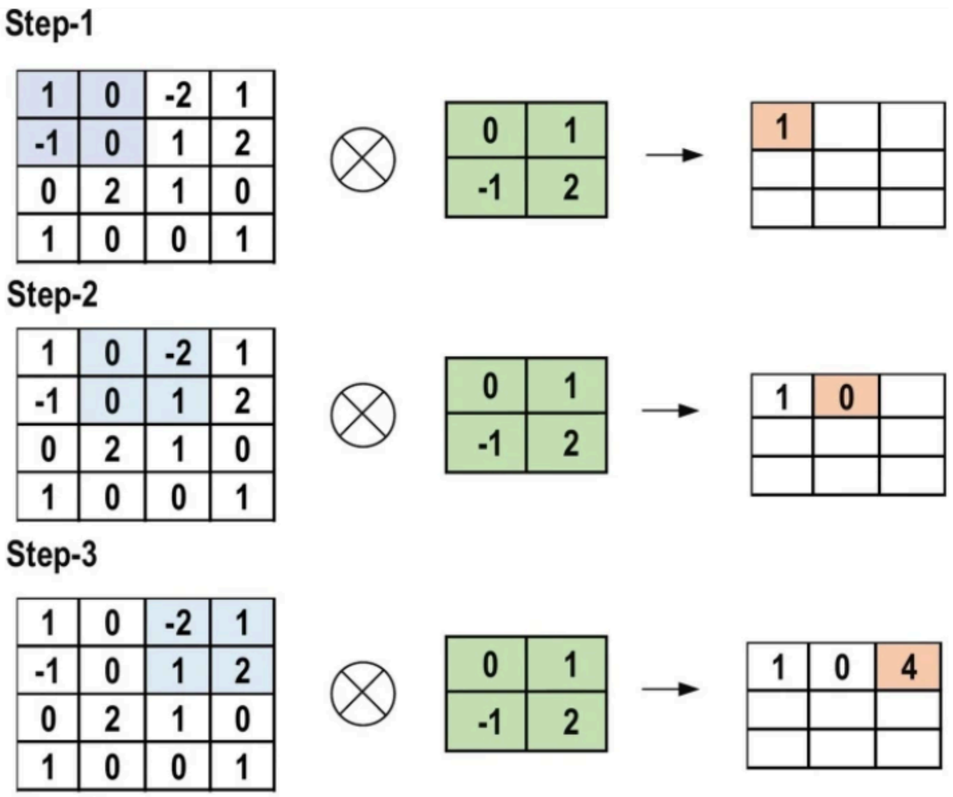


Figure 2: A visualization of the convolution operation using a 4x4 image and 2x2 kernel with stride = 1 and no padding. Adapted from "*Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*" (Alzubaidi et al., 2021).

As seen in Figure 2, many different patches are considered in the input image. We take the dot product of the patch from the input image and the kernel. Mathematically, this is summation of the entries of the matrix product of these 2 matrices. To introduce **non-linearity** and simulate an **action potential** in the human brain (Hodgkin & Huxley, 1952), we use the **Rectified Linear Unit (ReLU)** activation function.

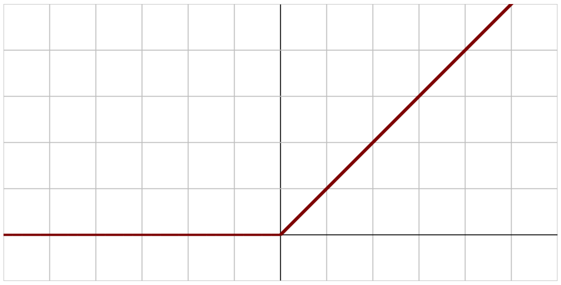
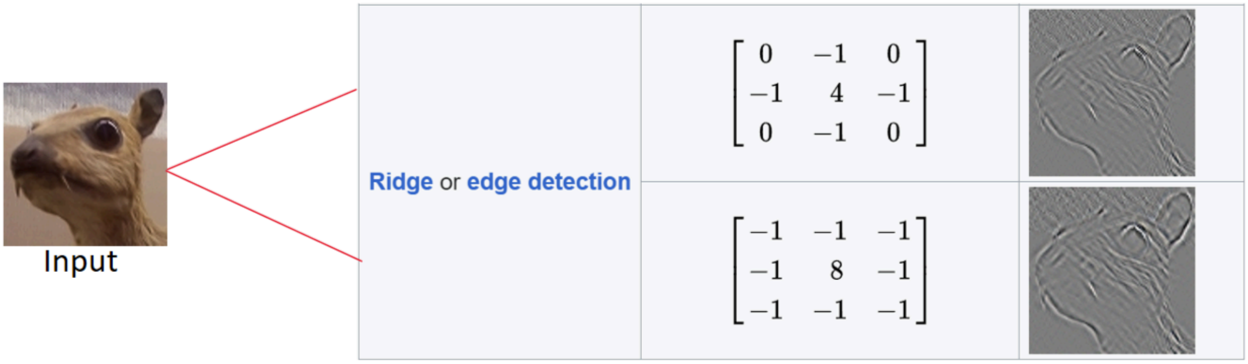


Figure 3: Graph of the Rectified Linear Unit. The ReLU activation function introduces non-linearity in a manner similar to the human brain. Adopted from Wikipedia (Activation Function).

As previously noted, this operation is done over the entire image and the resulting scalars at each step are stored in a smaller matrix called the **feature map**. It should be noted that specific kernels extract/”look for” different things (Figure 4).



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Figure 4: Examples of ridge or edge detection kernels and the resulting feature maps (right). Adapted from Wikipedia’s “Kernel (Image Processing)” (2023).

At the first layer, the kernels used result in low-level, high-contrast regions such as corners and edges. In deeper layers, the lower layer features are combined to yield more complex, higher level features.

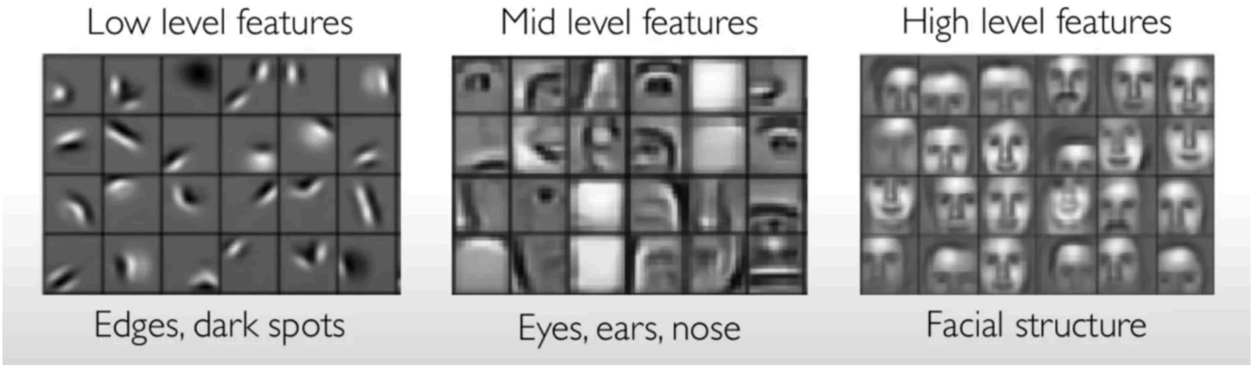


Figure 5: Example of hierarchy of CNN features. Adapted from MIT 6.S191 “*Introduction to Deep Learning”* (2024).

The Standard CNN Architecture

In the end, a standard CNN architecture takes an input image and applies the convolution operation with n_{1} kernels. This yields n_{1} feature maps. These first-level feature maps are representations of which features of the input image each kernel is picking up on. The feature maps are then typically pooled to decrease the dimension of the parameter space. A common pooling practice is max pooling on all non-intersecting 2x2 regions of the feature map. In max pooling for a 2x2 region, we map the highest value of the matrix to the subsampled layer.

These n_{1} pooled feature maps are then taken to be the input of the next convolutional layer. In this next convolutional layer, we have an input with n_{1} channels. If we wish to use n_{2} kernels at this layer, we must actually learn $n_{1} x n_{2}$ kernels in order to take each channel into account. This nonetheless leaves us with n_{2} features maps at the second layer due to an addition operation.

We again perform max pooling and this time end with n_{2} pooled feature maps. We can add more and more convolutional layers, repeating these same steps. In the end, we will have k feature maps. We flatten these k matrices into a vector that will feed into a standard feed-forward architecture which aims to classify the input.

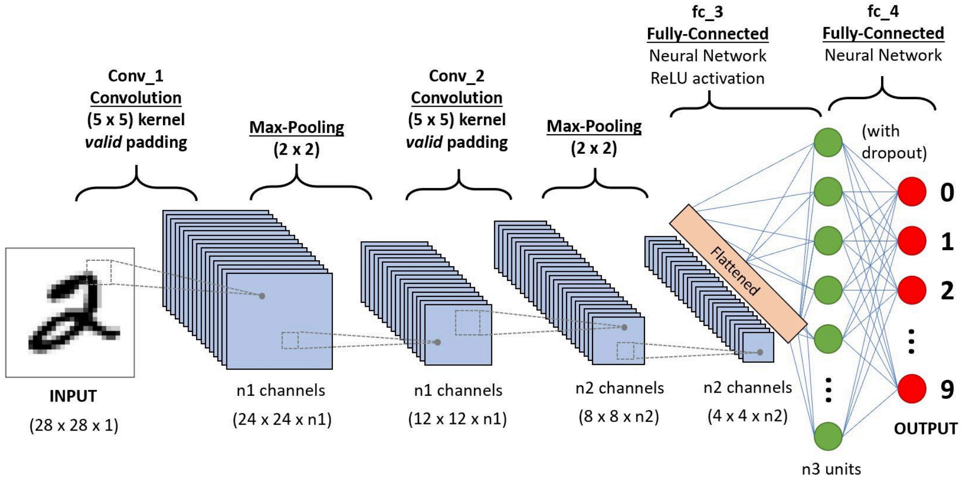
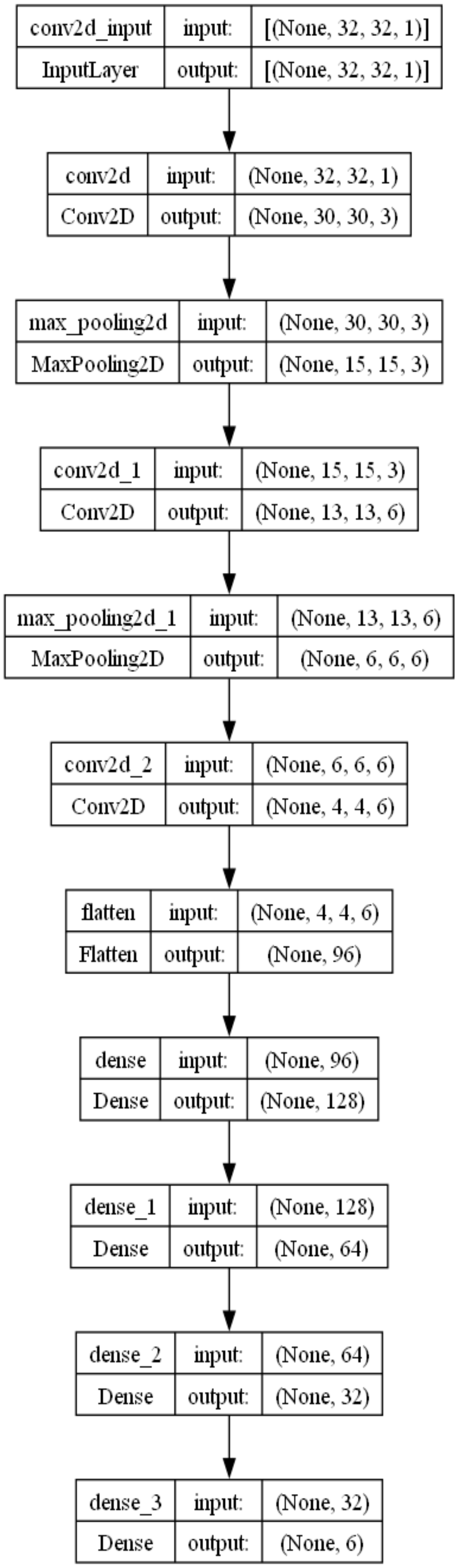


Figure 6: Standard CNN architecture. Adapted from "*What is the Convolutional Neural Network Architecture?*" (Ratan, 2023). https://www.analyticsvidhya.com/blog/2020/10/what-is-the-convolutional-neural-network-architecture/

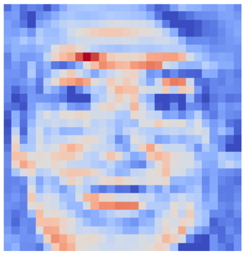
The weights and bias terms used for all kernels, as well for the dense portion are found according to empirical risk minimization. In the case of multiple classification, a standard framework is to use stochastic gradient descent, optimizing the categorical cross-entropy loss.

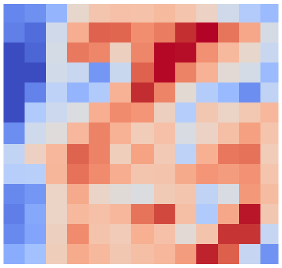
Facial Recognition

In this section, we use a CNN to classify actresses/actors from the facescrub dataset. In particular, given an input image of either Gerard Butler, Daniel Radcliffe, Michael Vartan, Lorraine Bracco, Peri Gilpin, or Angie Harmon, we wish to classify which person is in the image. The training test consists of 80% of all images of each actor/actress and the testing set contains the remaining 20% of images for each actor/actress. For the training set, there are 457 normalized (unit interval) images (grayscale and downsampled to 32x32).


conv2d_input | input: | [(None, 32, 32, 1)]
InputLayer | output: | [(None, 32, 32, 1)]
conv2d | input: | (None, 32, 32, 1)
Conv2D | output: | (None, 30, 30, 3)
max_pooling2d | input: | (None, 30, 30, 3)
MaxPooling2D | output: | (None, 15, 15, 3)
conv2d_1 | input: | (None, 15, 15, 3)
Conv2D | output: | (None, 13, 13, 6)
max_pooling2d_1 | input: | (None, 13, 13, 6)
MaxPooling2D | output: | (None, 6, 6, 6)
conv2d_2 | input: | (None, 6, 6, 6)
Conv2D | output: | (None, 4, 4, 6)
flatten | input: | (None, 4, 4, 6)
Flatten | output: | (None, 96)
dense | input: | (None, 96)
Dense | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 64)
dense_2 | input: | (None, 64)
Dense | output: | (None, 32)
dense_3 | input: | (None, 32)
Dense | output: | (None, 6)

The trained model uses three **3x3 kernels** in the first convolutional layer. In the second and third convolutional layers, 6 kernels are used. **ReLU** is used everywhere except for the output layer. No padding is used. For each convolutional layer, a stride of 1 is used. At the end of the first and second convolutional layers, **max pooling using a 2x2 submatrix** is used. At the end of the third convolutional layer, the 6 feature maps are flattened into a column vector in **$\mathbb{R}^96$**. This columihjn vector functions as the input for the dense layer (also using ReLU). In the end, a **softmax layer** classifies the image. The model is optimized through **Adam** so as to minimize the **categorical cross-entropy** loss function. After **128 epochs,** the **training accuracy was $/approx 100%$**. The resulting **testing accuracy was 86.02%.**

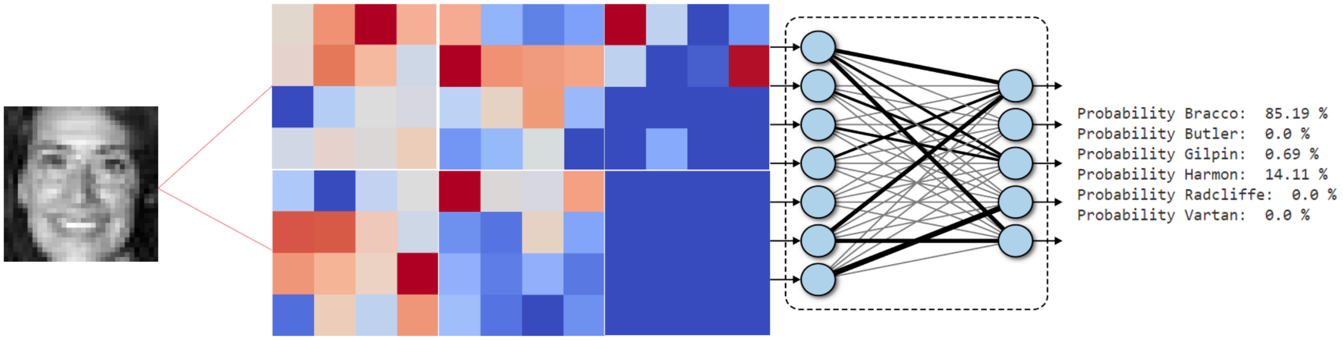




The low-level feature maps pick up on sharp, high-contrast regions such as sections of Lorraine Bracco's jaw, eyebrows, and earlobes (regions in red). This is to be expected as it is analogous to a human child's first month of vision. In comparison, we see more complex activations in the second convolutional layer of the CNN. In particular, the image shown clearly displays a composition of lower level features such as the regions near Lorraine Bracco's forehead, nose, and brow ridge. These are standard features appearing on a human face that allow us to recognize who a person is with remarkable accuracy. For instance, the reader may have seen cute online games that involve guessing who a celebrity is based solely on their eyes and been shocked by their accuracy.

Figure 7: Leonardo DiCaprio's eyes. Adapted from *"Using Only Their Eyes, Can You Correctly Guess The Names Of These Celebrities?"* (2022). buzzfeed.com

The high-level features found in the third convolutional layers are, as to be expected, a further combination of lower level features to yield more complex features. However, due to the convolution and pooling operations reducing dimensionality, they are more difficult to interpret to a human. In the end, for each input image, we end with 6 high-level feature maps. These are to be flattened, fed through a standard fully-connected feed-forward neural network, and then through a softmax layer for classification.

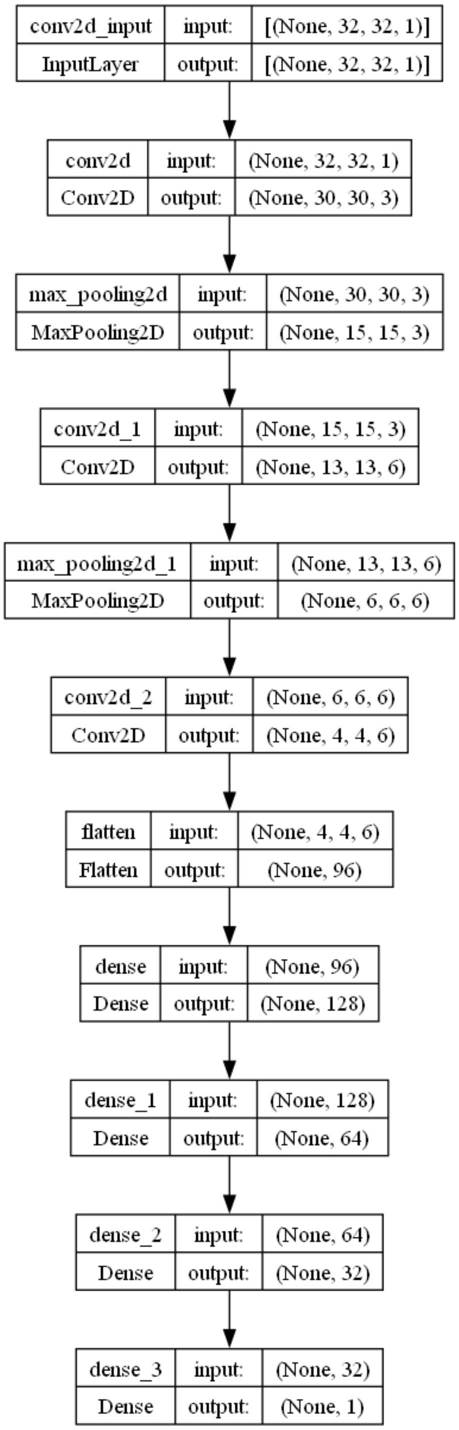


```
Probability Bracco:  85.19 %
Probability Butler:  0.0 %
Probability Gilpin:  0.69 %
Probability Harmon:  14.11 %
Probability Radcliffe:  0.0 %
Probability Vartan:  0.0 %
```

Figure 8: A depiction of the resulting feature maps to be flattened from an input image of Lorraine Bracco.

Gender Classification

In this section, we train a CNN to classify the gender of actors/actresses. The model is trained on 457 normalized (unit interval) images of Gerard Butler, Daniel Radcliffe, Michael Vartan, Lorraine Bracco, Peri Gilpin, or Angie Harmon (grayscale and downsampled to 32x32). The training test consists of 80% of all male and female images and the testing set contains the remaining 20% of images.


conv2d_input | input: | [(None, 32, 32, 1)]
InputLayer | output: | [(None, 32, 32, 1)]
conv2d | input: | (None, 32, 32, 1)
Conv2D | output: | (None, 30, 30, 3)
max_pooling2d | input: | (None, 30, 30, 3)
MaxPooling2D | output: | (None, 15, 15, 3)
conv2d_1 | input: | (None, 15, 15, 3)
Conv2D | output: | (None, 13, 13, 6)
max_pooling2d_1 | input: | (None, 13, 13, 6)
MaxPooling2D | output: | (None, 6, 6, 6)
conv2d_2 | input: | (None, 6, 6, 6)
Conv2D | output: | (None, 4, 4, 6)
flatten | input: | (None, 4, 4, 6)
Flatten | output: | (None, 96)
dense | input: | (None, 96)
Dense | output: | (None, 128)
dense_1 | input: | (None, 128)
Dense | output: | (None, 64)
dense_2 | input: | (None, 64)
Dense | output: | (None, 32)
dense_3 | input: | (None, 32)
Dense | output: | (None, 1)

The model architecture is nearly identical to the CNN used for facial recognition, except that it uses a sigmoid output layer for binary classification as opposed to a softmax layer. The following structure was used:

- Three 3x3 kernels in the first convolutional layer
- Six 3x3 kernels in the second convolutional layer
- Six 3x3 kernels in the third convolutional layer
- ReLU activation used in all layers except output
- Stride of 1 was used
- 2x2 max pooling used between conv. Layers
- No padding
- Adam optimizer
- Loss functional: Binary Crossentropy

After **64 epochs,** the **training accuracy was $/approx 100%$**. The resulting **testing accuracy was 90.22%.**

Another testing run was done in which the same CNN trained on Gerard Butler, Daniel Radcliffe, Michael Vartan, Lorraine Bracco, Peri Gilpin, or Angie Harmon was used to classify the genders of 24 actresses/actors it had **not prior been exposed to**. This **reduced testing accuracy down to 80.43%.** Testing the pre-trained model on people the model has not prior seen qualitatively appeared to yield feature maps with lower activations. This is likely explained by the fact that the feature detectors were not trained on these people and hence when they were exposed to different looking jaws, eyes, noses, etc. they had a hard time picking up on them.

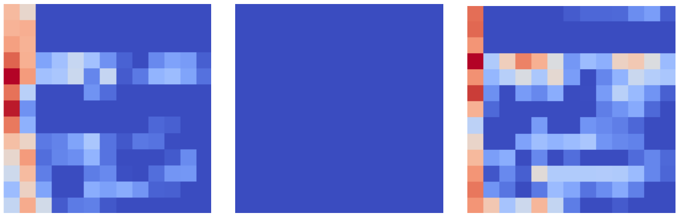
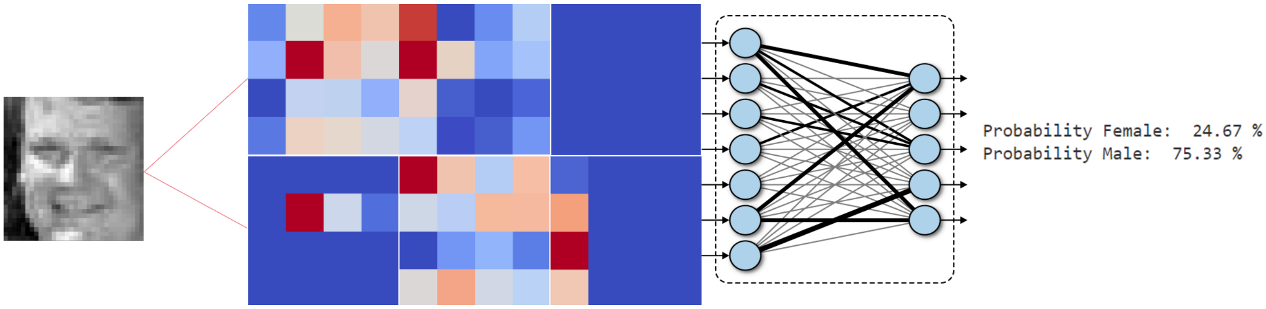


Figure 9: Feature Maps with minimal activations from the second convolutional layer.

Consequently, after the third convolutional layer, the feature maps to be flattened and inputted into a dense network had quite sparse activations, relative to the model on facial recognition which used the same people for both training and testing.



```
Probability Female:  24.67 %
Probability Male:  75.33 %
```

Figure 10: A depiction of the resulting feature maps to be flattened from an input image of Lorraine Bracco.

Conclusion

In sum, the CNN model performed very well on the dataset for both facial recognition and gender classification. For facial recognition, an 86.02% testing accuracy was obtained. Given the constraints of working with a small (457 training images) low-dimensional (32x32) dataset, this was quite remarkable. For gender classification (using the same people for both training and testing), a 90.22% was obtained, again quite good. When the testing set was changed to contain 1681 images of 24 people the model had not seen in training, the testing accuracy went down to 80.43%. These results could likely be improved if images were in higher resolution, lighting was held constant across the images, and if facial orientation was held constant (the same profile used in each image).

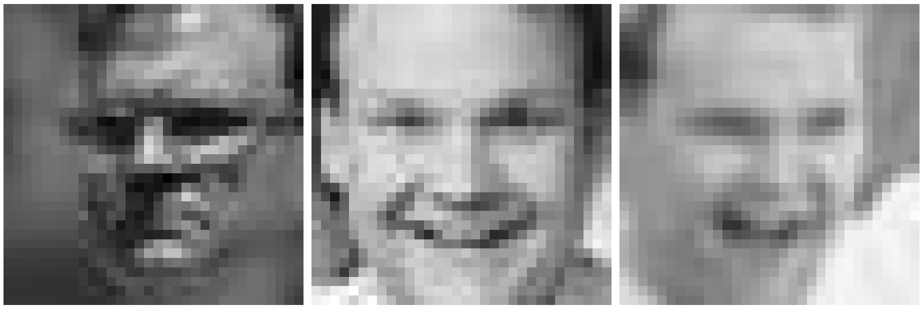


Figure 11: Images of Andy Richter which depict the variation seen in lighting and facial orientation seen in the dataset.

CNN-KNN Hybrid Model

In this section, we examine the use of CNN as a feature detection and dimensionality-reduction tool. In particular, note that a flattened 32x32 grayscale image can be represented as a vector in $\mathbb{R}^{1024}$. The CNN models used for facial and gender recognition reduce the 32x32 input image down into a vector in $\mathbb{R}^{96}$ prior to being inputted into a dense layer, more than one full order of magnitude lower dimension! In addition, these convolutional layers are trained to pick up on features that aid in the classification task, providing merit for their use as feature detectors.

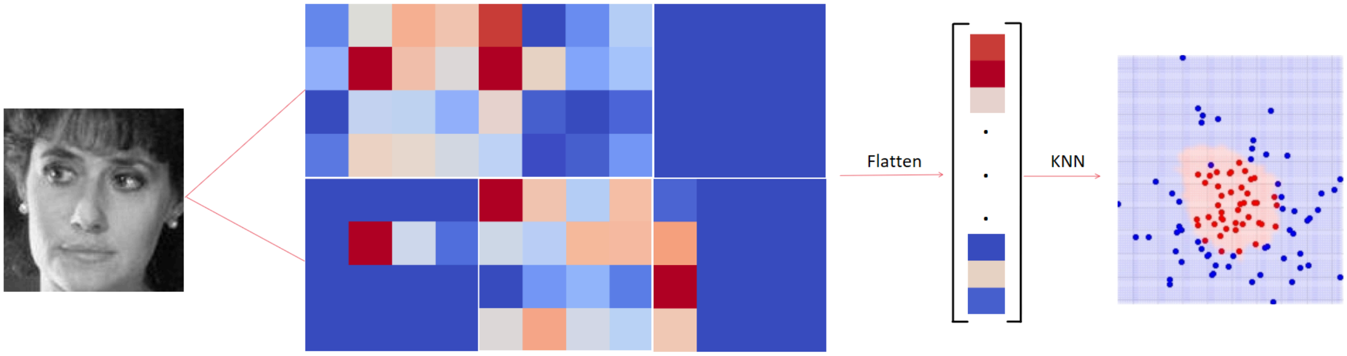


Figure 12: Overview of the CNN-KNN Hybrid Model.

Facial Recognition

The same CNN for facial recognition was used but the 6 feature maps were flattened into $\mathbb[R]^{96}$. Identical training and testing sets were used. GridSearchCV was performed with 5 folds. This yielded a training accuracy of 70.60% for optimal K=1 and a testing accuracy of 77.42%.

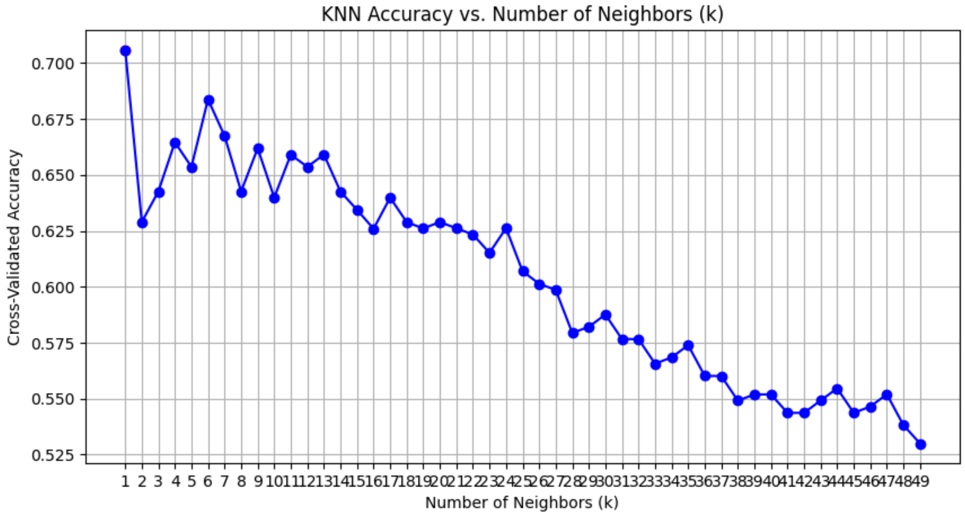


Figure 13: GridSearchCV results for 5 folds on the facial recognition task.

In comparison to the 86.02% testing accuracy achieved by the full CNN, this is a substantial decrease in accuracy yet it still maintains significantly higher accuracy than standard KNN.

Gender Classification

The same CNN for gender classification was used but the 6 feature maps were flattened into $\mathbb[R}^{96}$. Identical training and testing sets were used. GridSearchCV was performed with 5 folds. This yielded a training accuracy of 95.07% for optimal K=8 and a testing accuracy of 89.13%.

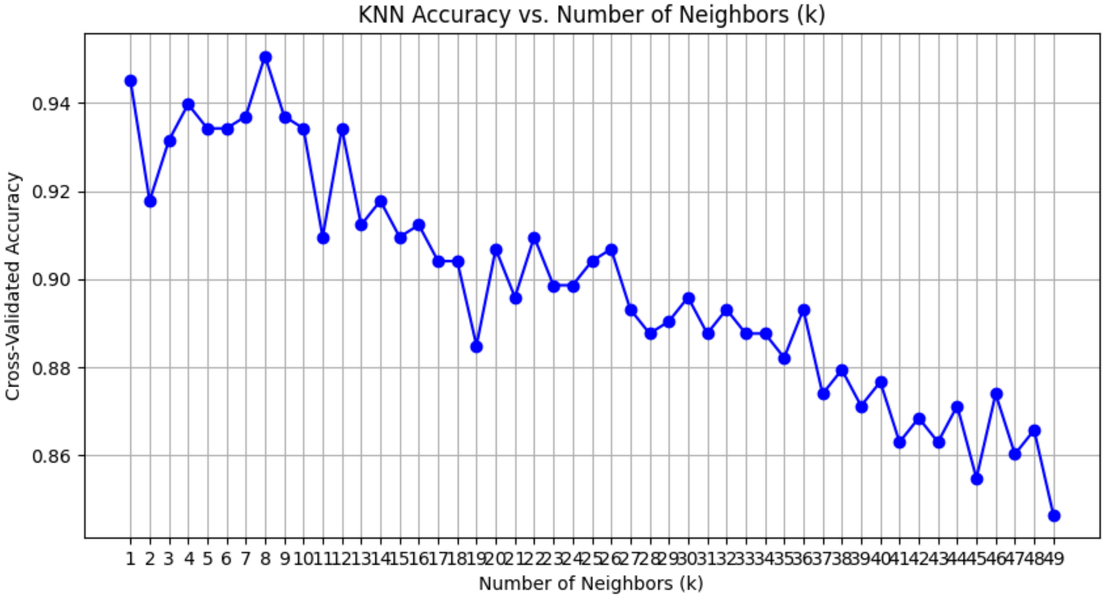


Figure 14: GridSearchCV results for 5 folds on the gender classification task.

In comparison to the 90.22% testing accuracy achieved by the full CNN, this is a near identical testing accuracy that still significantly outperforms KNN.

In maintaining the same training set, but altering the testing set to contain images of 24 people the model had not been prior exposed to, the training accuracy results are identical with K=8 being optimal (to be expected, since no changes were made), however the testing accuracy drops down to 76.50%. Compared to the full CNN which obtained 80.43% testing accuracy on this task, this is a noticeable decrease in performance. This finding may point to CNNs being a more robust framework than KNN.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J. *et al.* Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *J Big Data* 8, 53 (2021). https://doi.org/10.1186/s40537-021-00444-8

Hubel, D. H., Wiesel, T. N., (1962), Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160 doi: 10.1113/jphysiol.1962.sp006837.

Younger, A., Adler, S. A., & Vasta, R. (2012). Child Psychology: A Canadian Perspective (3rd ed.). Wiley.

Salapatek, P. (1968). Visual scanning of geometric figures by the human newborn. Journal of Comparative and Physiological Psychology, 66(2), 247–258. https://doi.org/10.1037/h0026376

HODGKIN AL, HUXLEY AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol. 1952 Aug;117(4):500-44. doi: 10.1113/jphysiol.1952.sp004764. PMID: 12991237; PMCID: PMC1392413.