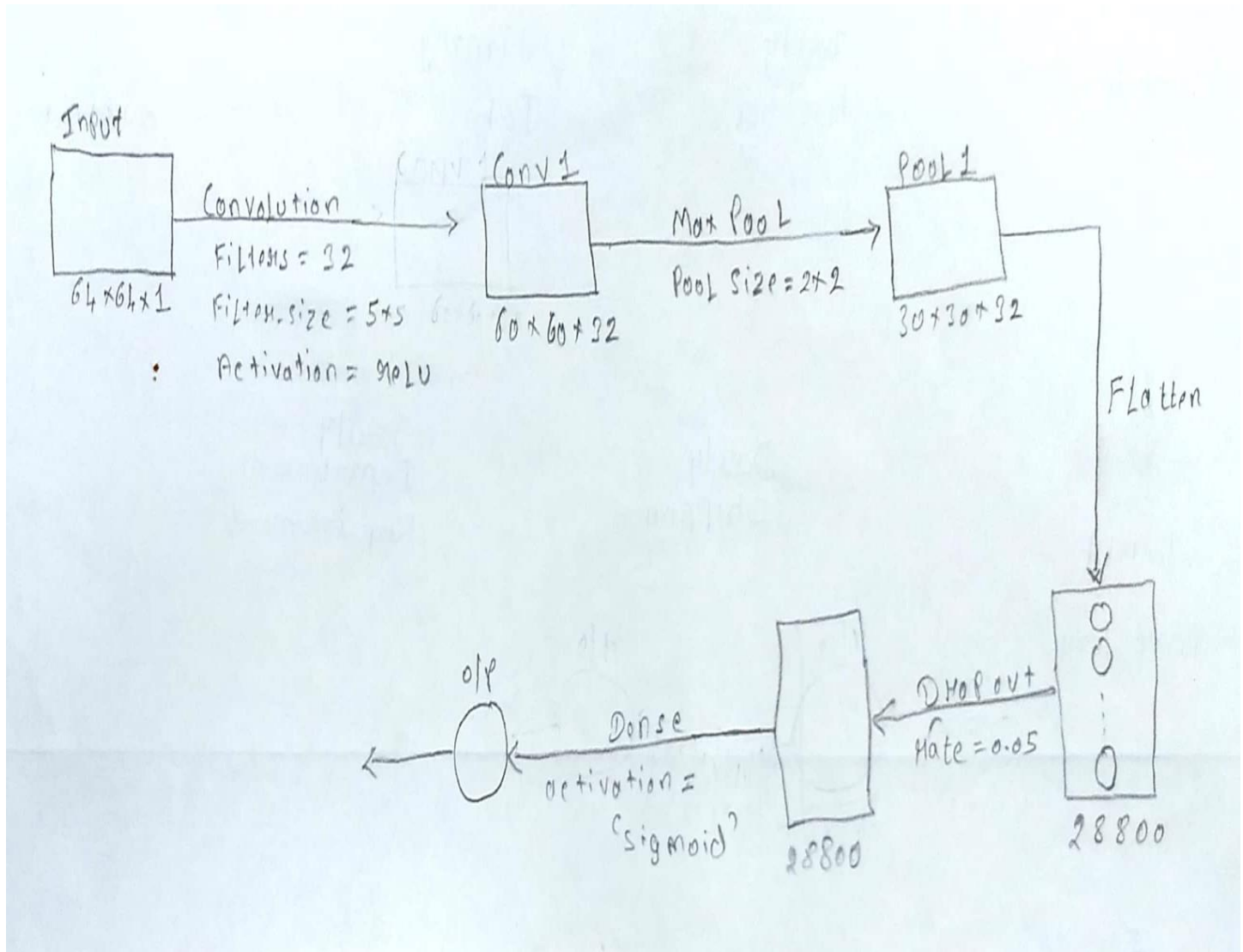
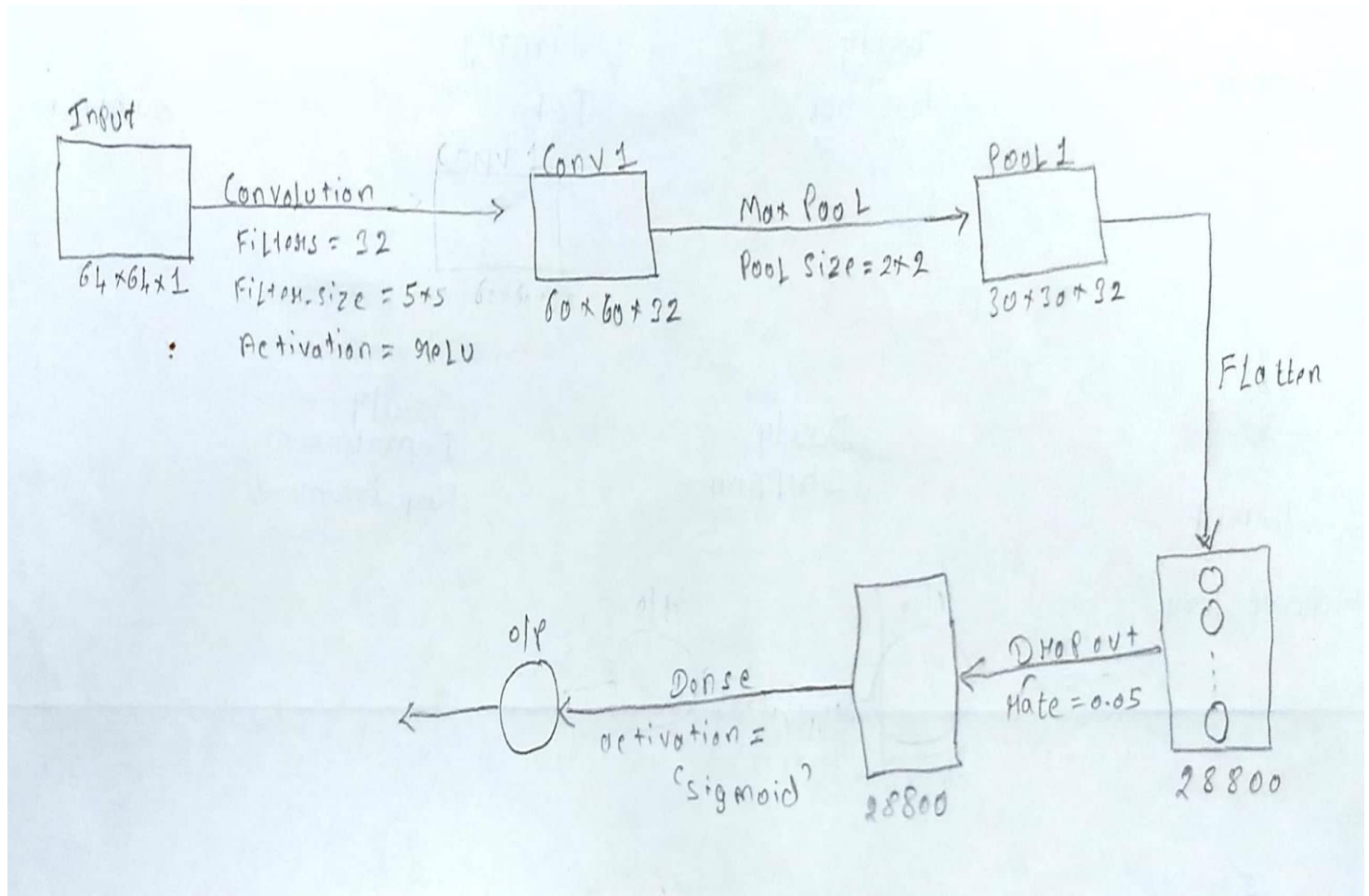


1. Bush:

- The above architecture which shows the Bush model with different layers showing their dimensions along with input and output size.
- It also shows various activation function used at different type of layers.
- My model has following details:
  - Input:
    - $64 \times 64 \times 1$
  - Convolution layer:
    - Filters = 32
    - Kernel size =  $5 \times 5$

- Activation = relu
    - Output = 60x60x32
  - Max Pooling:
    - Pool size = 2x2
    - Output = 30x30x32
  - Flatten:
    - Output = 1D – 28800
  - Dropout:
    - Rate = 0.05
    - Output: 1D – 28800
  - Output Layer:
    - Dense
    - Activation = sigmoid
- The f1 score that I could achieve was:
- Train data: 0.9800000000000001
  - Test data: 0.7761194029850746

2. Williams:

- The above architecture which shows the Williams model with different layers showing their dimensions along with input and output size.
- It also shows various activation function used at different type of layers.
- My model has following details:
  - Input:
    - $64 \times 64 \times 1$
  - Convolution layer:
    - Filters = 32
    - Kernel size =  $5 \times 5$
    - Activation = relu
    - Output =  $60 \times 60 \times 32$
  - Max Pooling:
    - Pool size =  $2 \times 2$

- Output = 30x30x32
- Flatten:
  - Output = 1D – 28800
- Dropout:
  - Rate = 0.05
  - Output: 1D – 28800
- Output Layer:
  - Dense
  - Activation = sigmoid
- The f1 score that I could achieve was:
  - Train data: 0.8923076923076922
  - Test data: 0.7096774193548386
  
- The number of epochs I varied was from 3 to 15 for both of them. I got good result around 10 or 11 but the result kept changing as I re-run my program.
- I have used simple model as the dataset size is not that sufficient and if I use complex model, it may overfit the data.
- The f1 score for Bush is more comparatively to Williams as there are more examples for Bush in training set as compared to Williams and hence better training and prediction on the data which ultimately leads to more f1-score.