

# Hackathon: Fast Fashion

Group 2

Daniel Wang, Kate Skibo,  
Nicholas Nguyen, Steve Shields



# Table of contents

## 01 Problem Statement

Not from this world

## 02 The Data

Fashion-MNIST

## 03 Modeling

Neural Networks

## 04 Parameter Tuning

## 05 Comparison

A look at the models

## 06 Conclusion

Which was the best?



# 01 Problem Statement

# Problem Statement

- **Shapeshifting** militant **insurgents** from the planet Omicron-6
- **Enemy leader** meeting backers at **Fashion Week** on alpha-Earth in disguise
- **Must learn human fashion terms to blend in** at Fashion Week
- Infiltrate the events to **ASSASSINATE** THE (fake) PRIME MINISTER OF MALAYSIA!



# 02

## The Data

Fashion-MNIST dataset (from Kaggle)



# Fashion-MNIST:

Dataset consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.



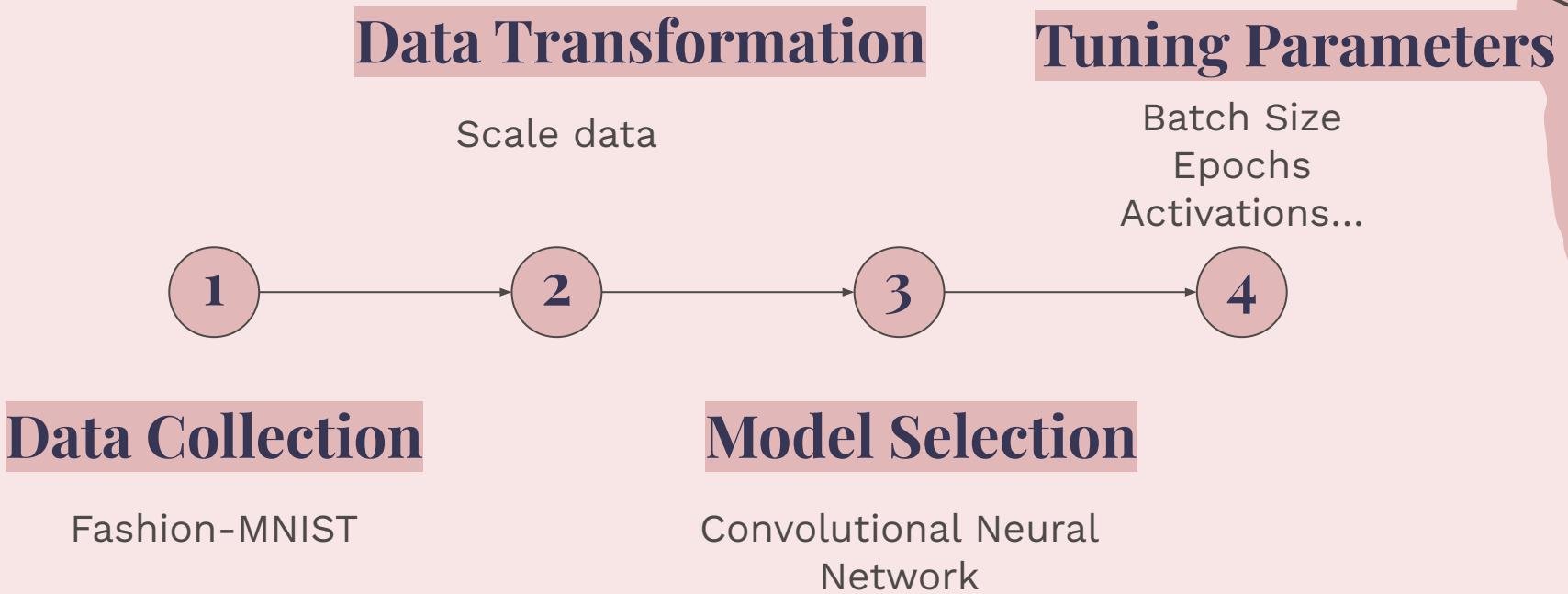
# Label Legend:

Training dataset :

- 0 = T-shirt/top
- 1 = Trouser
- 2 = Pullover
- 3 = Dress
- 4 = Coat
- 5 = Sandal
- 6 = Shirt
- 7 = Sneaker
- 8 = Bag
- 9 = Ankle boot



# Our process





03

# Model Selection

Convolutional Neural Networks

# Modeling Overview

## TensorFlow & Keras

Using Keras, we created a variety of sequential models, building layer by layer, trying different parameter values and the order and number of layers to find the best performing one.

In our base model...



# 04

## Parameter Selection

Layers, Filters, Kernel size, Batch size,  
Epochs, Optimizer Type

# Params:



## Layers

2 or 3



## Epochs

10



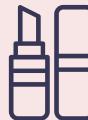
## Dropouts

Yes/No  
Where



## Flattening

Flatten after first layer



## Filters

32, 64, 128



## Optimizer

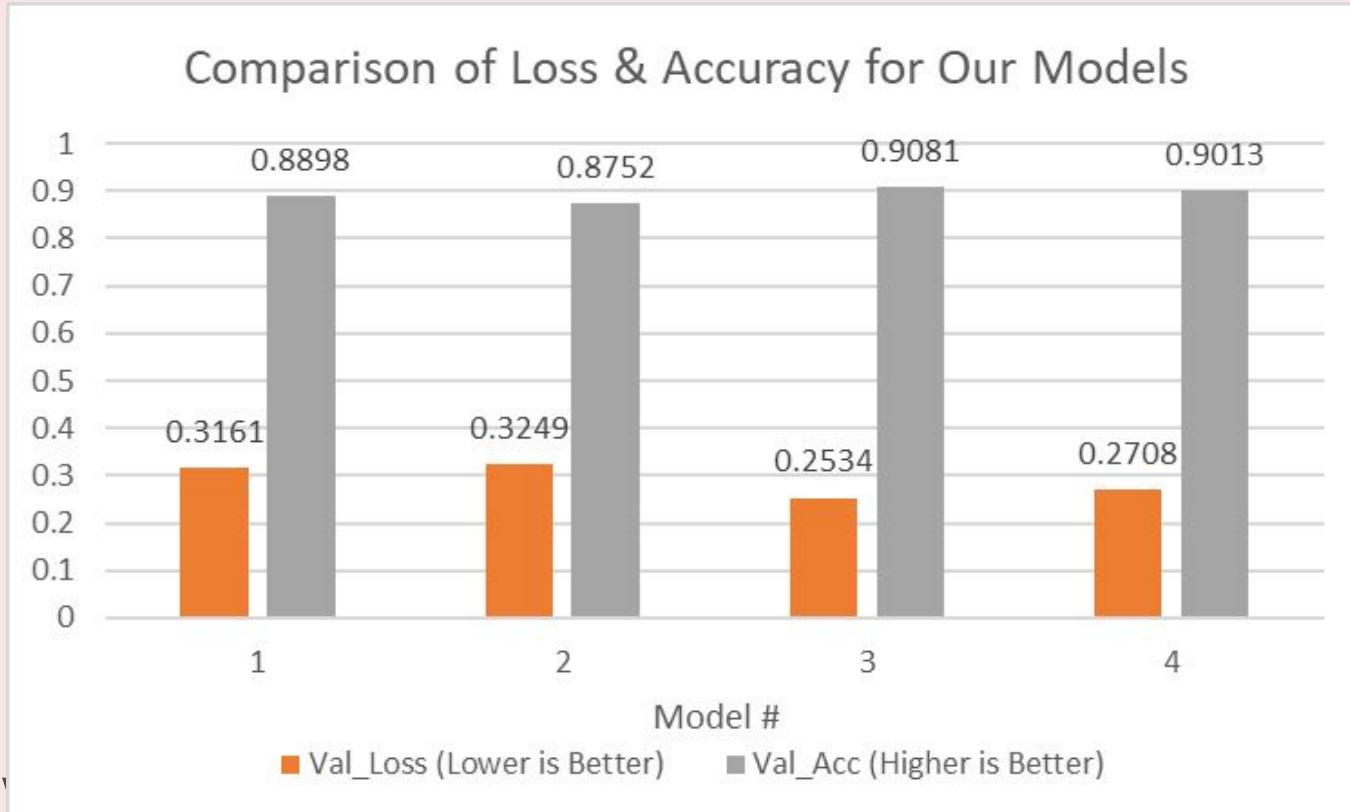
Rmsprop vs Adam

# 05

## Model Comparison



# All Models Summary



# All Models Summary

	Optimizer	Layers	Dropout	Loss	Accuracy	Val_Loss	Val_Acc
	1 rmsprop	2	No	0.286	0.9009	0.3161	0.8898
	2 rmsprop	2	Yes	0.3092	0.8876	0.3249	0.8752
	<b>3 rmsprop</b>	<b>3</b>	<b>Yes</b>	<b>0.1959</b>	<b>0.9273</b>	<b>0.2534</b>	<b>0.9081</b>
	4 Adam	3	Yes	0.1565	0.9429	0.2708	0.9013

# Measure of Success:

## Accuracy

How well the model performs across all classes.

Since we have 10 classes this will be the best metric to utilize.

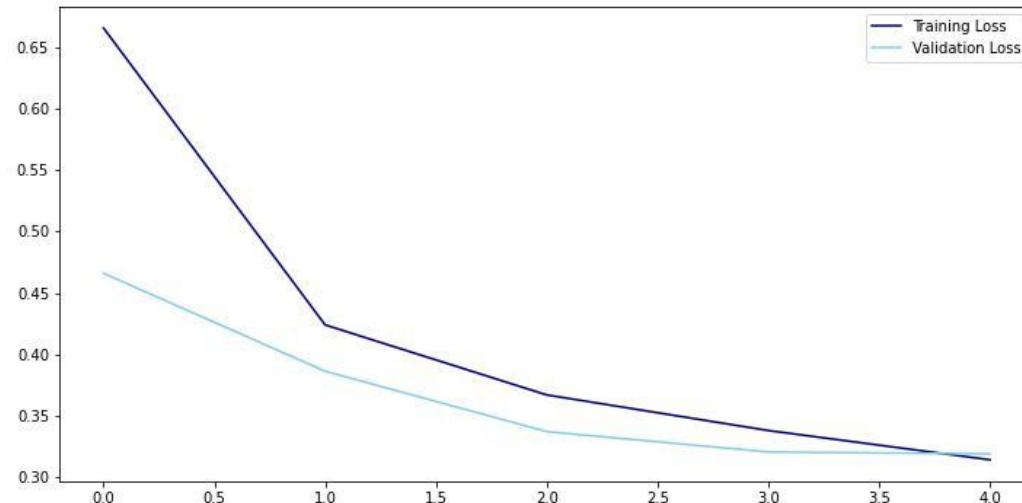
Also looking to minimize loss as much as possible without sacrificing accuracy.

Baseline Accuracy = 10%



# Model 1:

- 2 Convolutional layers (32, 64)
- 1 Dense Layer (10)
- Optimizer (rmsprop)

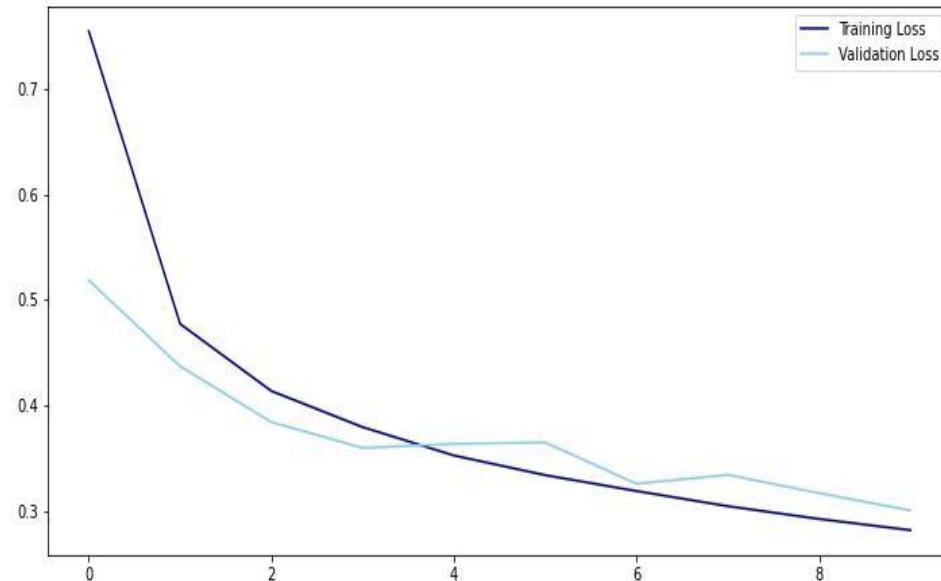


# Model 1:

	Optimizer	Layers	Dropout	Loss	Accuracy	Val_Loss	Val_Acc
✗	1 rmsprop	2	No	0.286	0.9009	0.3161	0.8898
✗	2 rmsprop	2	Yes	0.3092	0.8876	0.3249	0.8752
✓	3 rmsprop	2	Yes	0.1959	0.9273	0.2534	0.9081
✗	4 Adam	3	Yes	0.1565	0.9429	0.2708	0.9013

# Model 2:

- 2 Convolutional layers (32, 64)
- 1 Dense Layer (10)
- Optimizer (rmsprop)

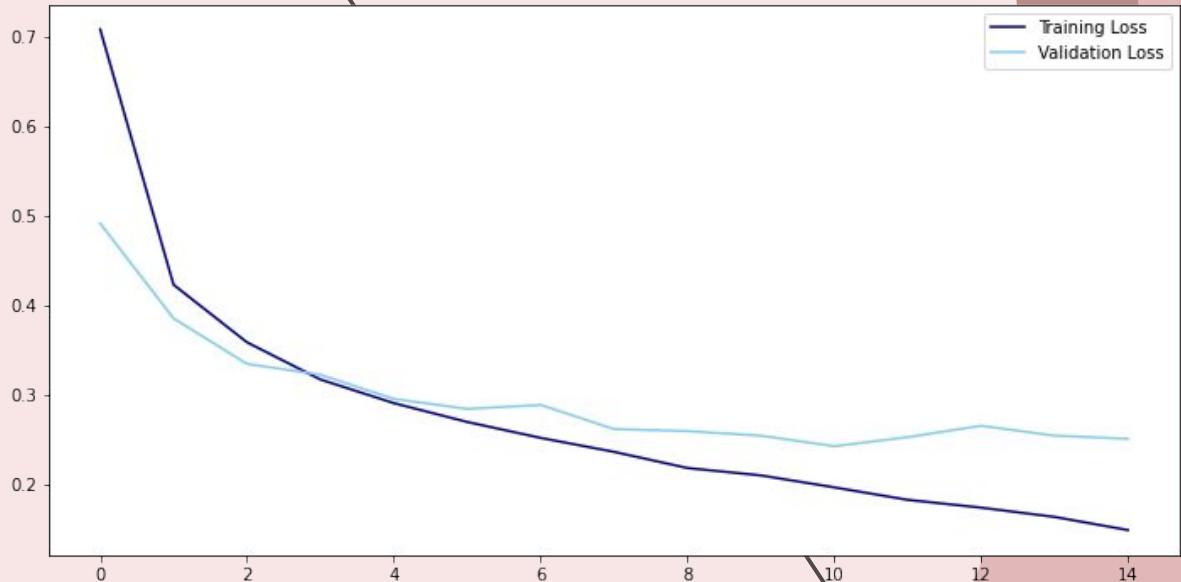


# Model 2:

	Optimizer	Layers	Dropout	Loss	Accuracy	Val_Loss	Val_Acc
✗	1 rmsprop	2	No	0.286	0.9009	0.3161	0.8898
✗	2 rmsprop	2	Yes	<b>0.3092</b>	<b>0.8876</b>	<b>0.3249</b>	<b>0.8752</b>
✓	3 rmsprop	2	Yes	0.1959	0.9273	0.2534	0.9081
✗	4 Adam	3	Yes	0.1565	0.9429	0.2708	0.9013

# Model 4:

- 3 Convolutional layers. Kernels 32, 64, 128
- 2 Dense layers, 128 & 10
- Optimizer (adam)

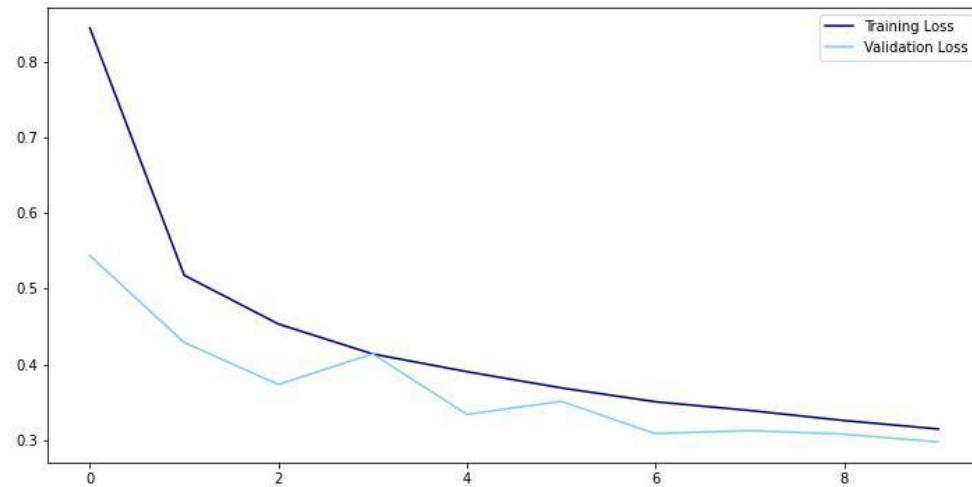


# Model 4:

	Optimizer	Layers	Dropout	Loss	Accuracy	Val_Loss	Val_Acc
✗	1 rmsprop	2	No	0.286	0.9009	0.3161	0.8898
✗	2 rmsprop	2	Yes	0.3092	0.8876	0.3249	0.8752
✓	3 rmsprop	2	Yes	0.1959	0.9273	0.2534	0.9081
✗	4 Adam	3	Yes	0.1565	0.9429	0.2708	0.9013

# Model 3:

- 2 Convolutional layers (32, 64)
- 2 Dense Layers (32, 10)
- Only model with **Dropout**
- Optimizer (rmsprop)



# Model 3:

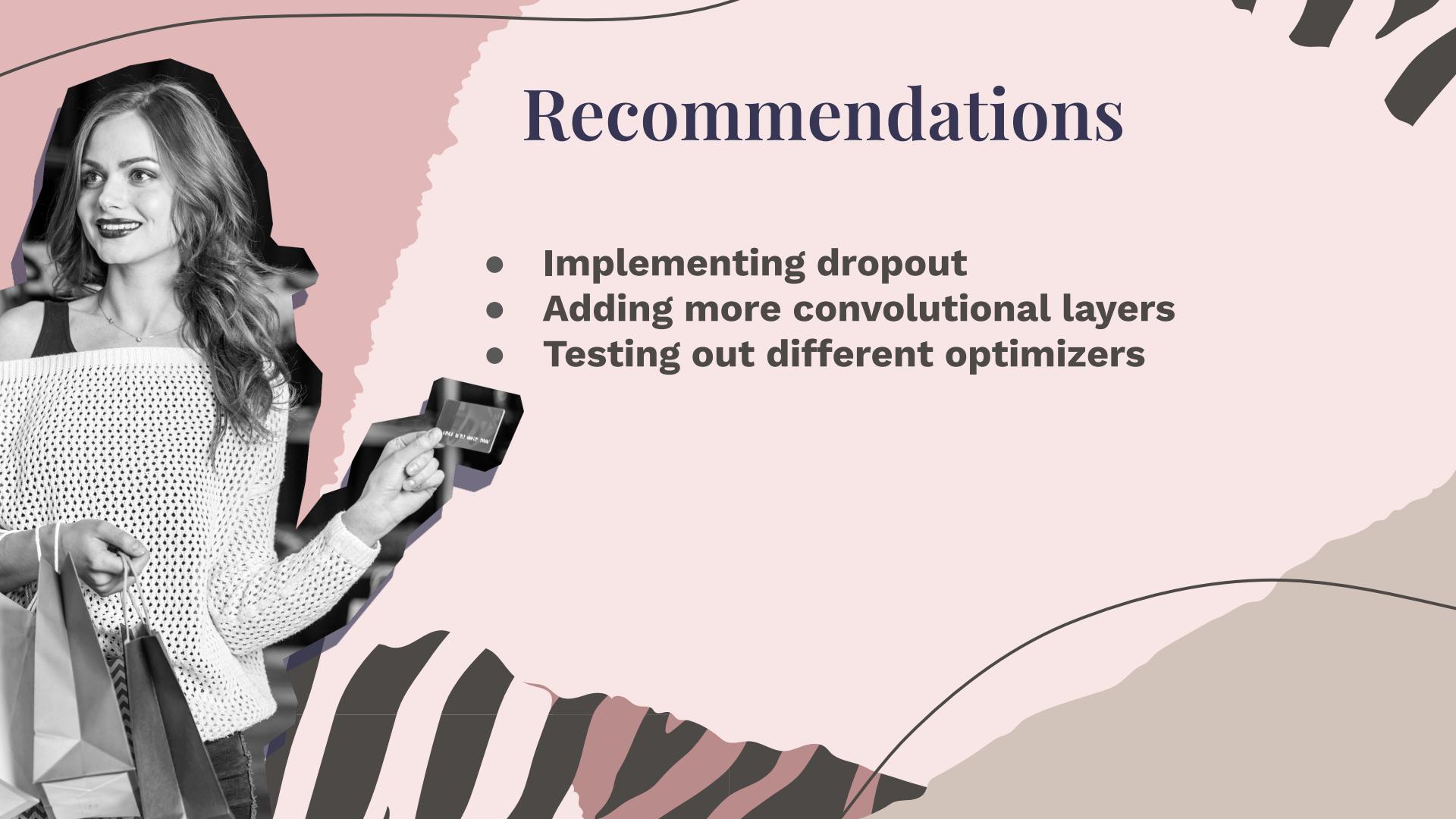
	Optimizer	Layers	Dropout	Loss	Accuracy	Val_Loss	Val_Acc
✗	1 rmsprop	2	No	0.286	0.9009	0.3161	0.8898
✗	2 rmsprop	2	Yes	0.3092	0.8876	0.3249	0.8752
✓	<b>3 rmsprop</b>	<b>2</b>	<b>Yes</b>	<b>0.1959</b>	<b>0.9273</b>	<b>0.2534</b>	<b>0.9081</b>
✗	4 Adam	3	Yes	0.1565	0.9429	0.2708	0.9013

# Conclusion

Model 3 performed the best with the validation loss significantly lower than the training loss.

Accuracy: 90.81

Validation Loss: 0.2534

A black and white photograph of a young woman with long, wavy hair, smiling at the camera. She is wearing a light-colored, patterned top and jeans. She is holding several shopping bags in one hand and a small electronic device, possibly a smartphone or a small tablet, in the other. The background is a soft, out-of-focus pink.

# Recommendations

- **Implementing dropout**
- **Adding more convolutional layers**
- **Testing out different optimizers**

# Epilogue

- Most common misclassification by the model is that it classifies 6:shirts as 0:T-shirt/top or 4:Coat
- Critical blunder! Operative blew his cover referring to a shirt as a coat.
- Enemy target escaped off-world.



# Thanks!

Do you have any questions?



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon** and infographics & images by **Freepik**

Please keep this slide for attribution

