Same Same or Different?

Finding duplicate products from Shopee listings

Background

E-commerce sites such as **Shopee** receive multiple product listings daily. To improve recommendations, there is a need to identify listings which represent the same products.

This will assist:

Sellers – with category recommendations to list products

Buyers – through recommendations of the same products (possibly cheaper) from other shops

Goal

With a given set of **product images**, determine which are the *same product*.

Product A

Which are the same as product A?

What is our Evaluation Metric?

Mean F1 Score

- Obtain F1 score for each product
- Get the mean F1 score for all the products in the dataset

$$\textbf{F1 Score} = \frac{\text{Precision x Sensitivity}}{\text{Precision + Sensitivity}}$$

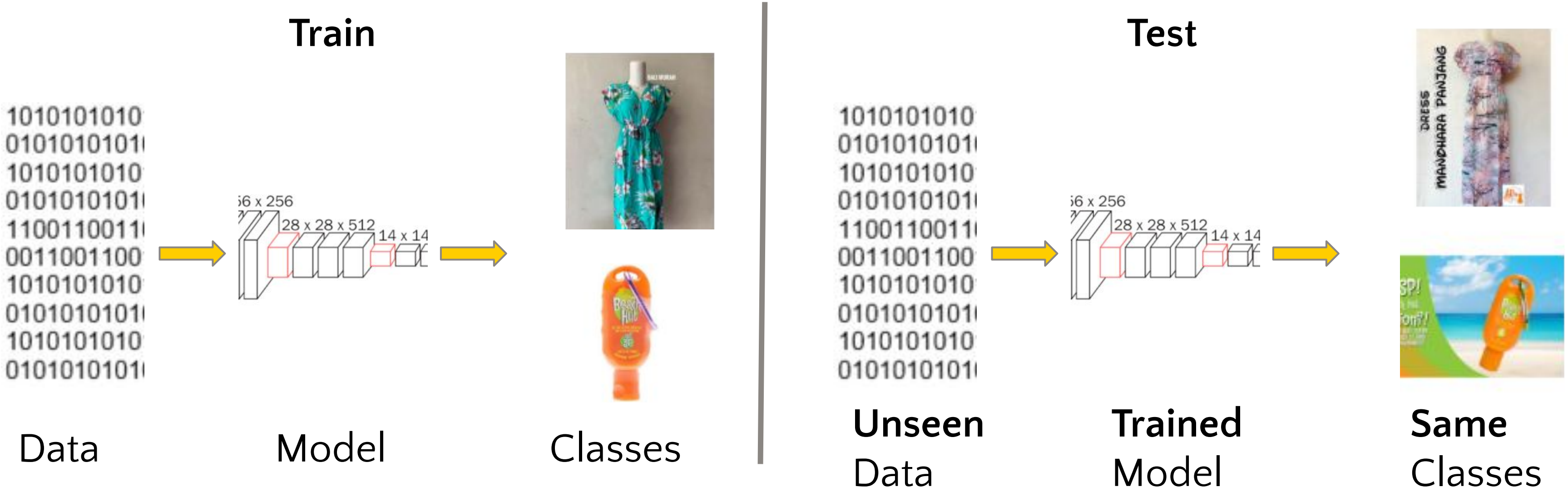
*range: 0 to 1

Contents

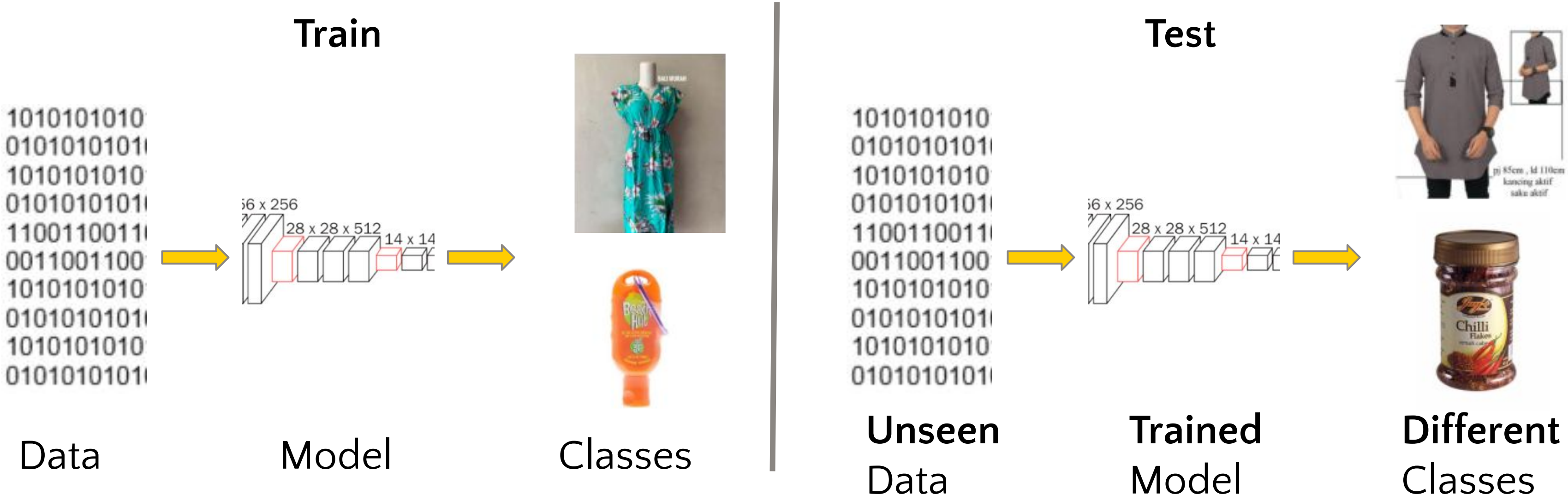
How to solve such problems?

How to solve such problems?

Typical multiclass classification



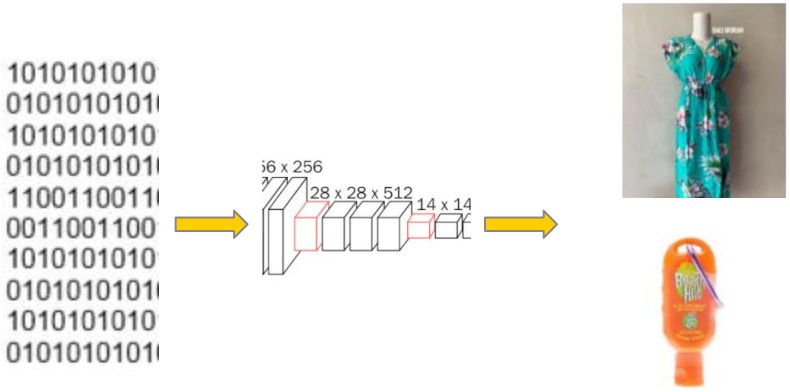
How to solve such problems?



How to solve such problems?

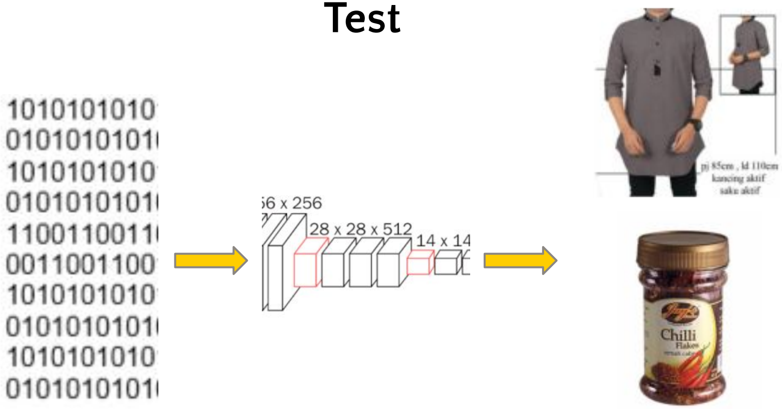
1. Model to pick up relevant features
2. Identify such features in new images

Train



Data Model Classes

Test



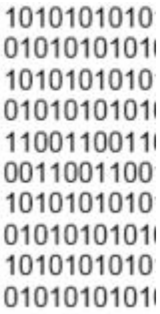
Unseen Data

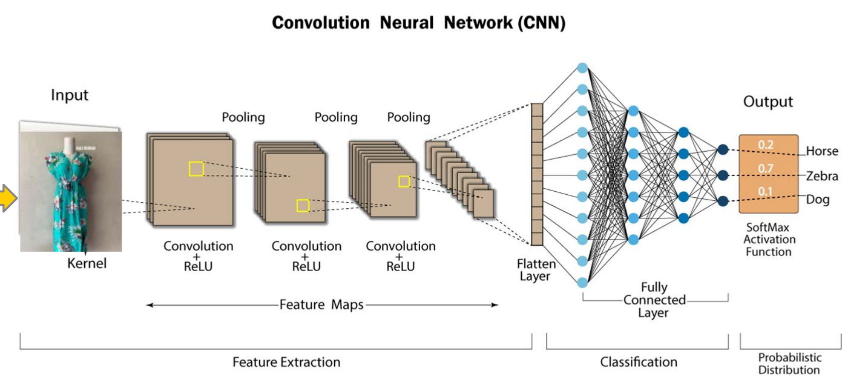
Trained Model

Different Classes

How to solve such problems?

1. Model to pick up relevant features



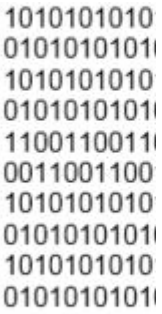


Data

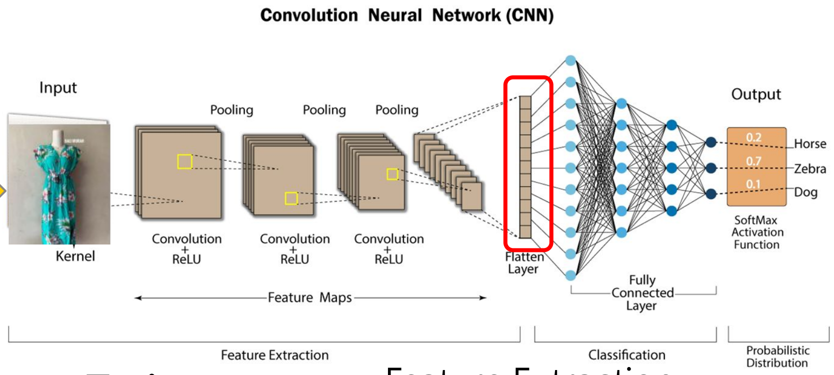
Train
Model

How to solve such problems?

1. **Model to pick up relevant features**



Data



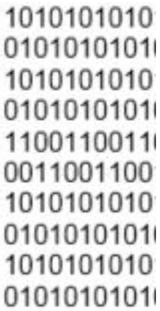
Train
Model

<u>Feature Extraction</u>
Extract the embeddings instead of the predictions.

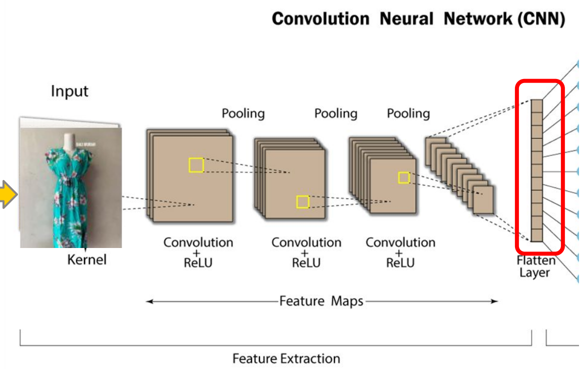
How to solve such problems?

1. Model to pick up relevant features

2. Identify such features in new images

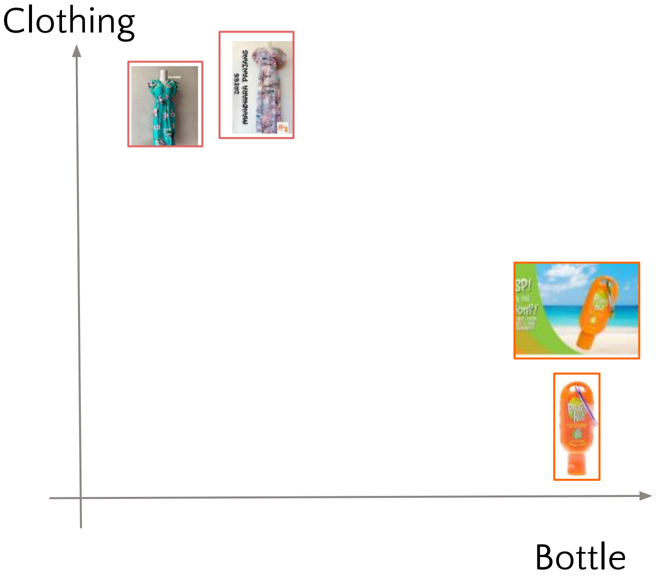


Data



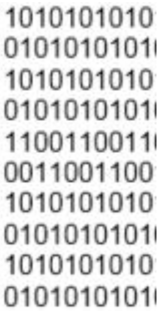
Train
Model

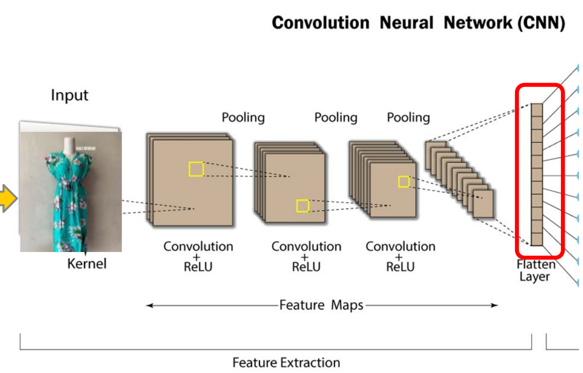
<u>Feature Extraction</u>
Extract the embeddings instead of the predictions.



How to solve such problems?

1. Model to pick up relevant features



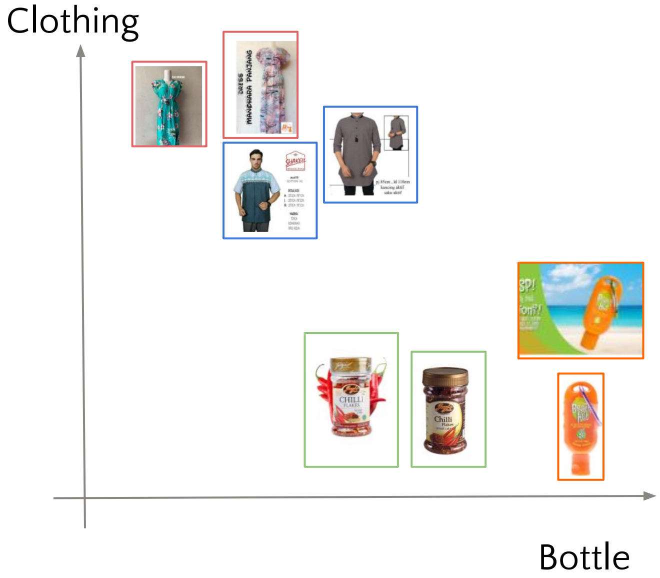


Data

Train
Model

<u>Feature Extraction</u>
Extract the embeddings instead of the predictions.

2. Identify such features in new images

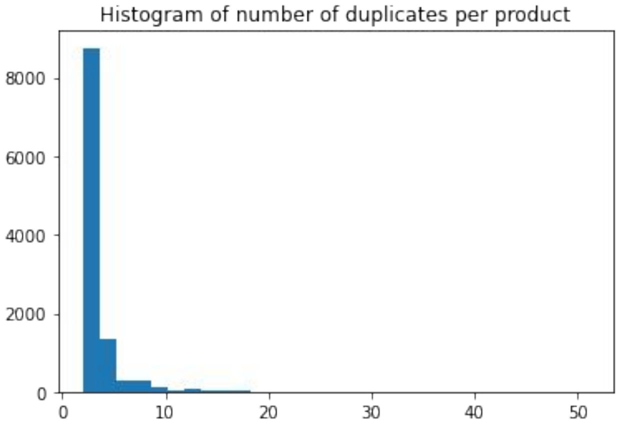


Exploratory Data Analysis

Frequency distribution of classes

Sample Size = 34,250

Number of Classes = 11,014

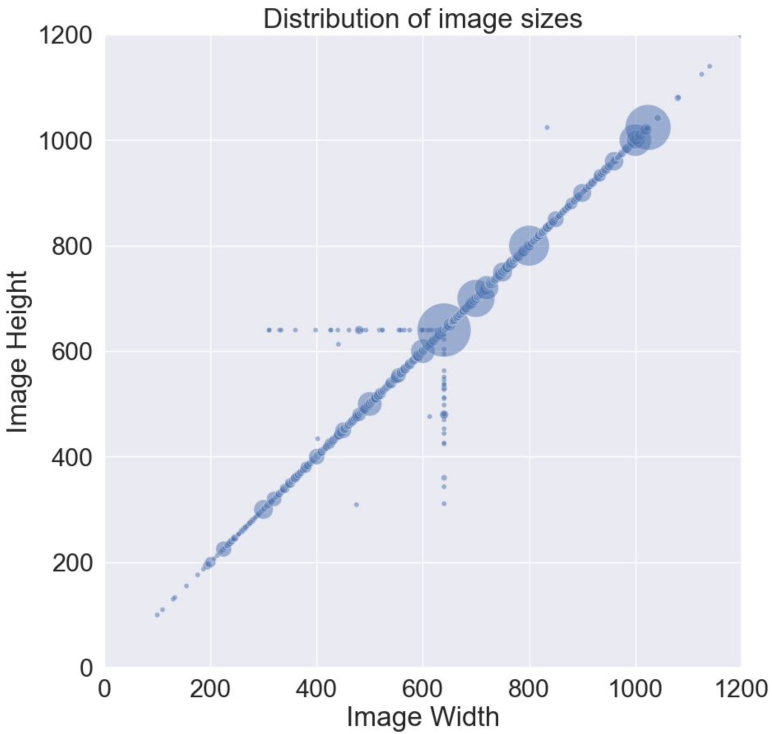


Number of products in the class	Number of classes
2	6979
3	1779
4	862
5	468
…	…
45	1
46	2
…	…

Distribution of image sizes

More than 99% of images are square shaped

Reading images into the model with square dimensions will not distort most images



What do the images look like?

Group A

Group B

Modelling

Modelling

Base Model
- phash difference

Choosing a pre-trained CNN
- VGG16
- ResNet50
- InceptionV3
- EfficientNetB0

Fine Tuning
- Diff layers unfrozen
- Diff Epochs trained

Metric
- Cosine Distance

Base Model

What is a Perceptual Hash?

A mathematical algorithm analyzes an image's content and represents it using a 64-bit number fingerprint.


```
array([[ True, False, False,  True, False,  True, False, False],
       [ True, False, False,  True, False,  True,  True,  True],
       [False,  True, False, False,  True,  True,  True,  True],
       [ True, False, False,  True, False, False,  True,  True],
       [False,  True,  True,  True,  True,  True, False,  True],
       [False,  True, False, False,  True,  True, False, False],
       [False, False,  True, False, False,  True, False, False],
       [False, False,  True,  True, False, False,  True,  True]])
```

Base Model (Phash)

Similar items, small phash difference

−

=

2*

Dissimilar items, large phash difference

−

=

20*

*these are simulated figures

Base Model (Phash)

Similar items, small phash difference

−

= 2*

Dissimilar items, large phash difference

−

= 20*

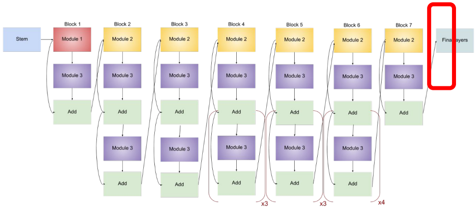
Model	Train (Mean F1)	Test (Mean F1)
phash	0.596	0.613

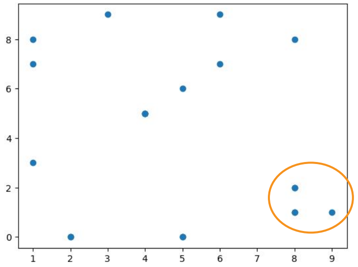
*these are simulated figures

Choosing a pre-trained CNN

Transfer Learning

If a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world

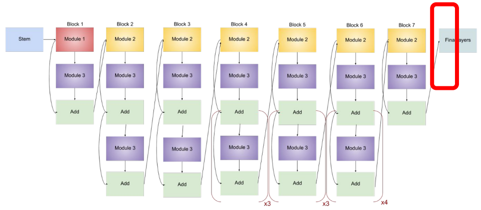


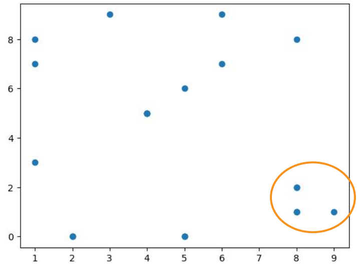


Choosing a pre-trained CNN

Transfer Learning

If a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world



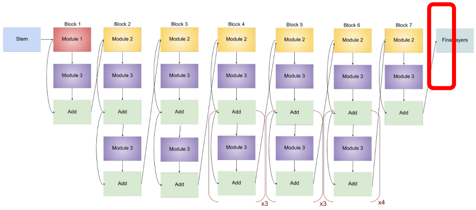


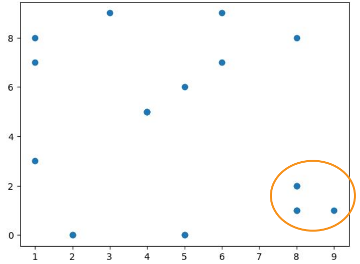
Model	Sample (Mean F1)
VGG16	0.588
ResNet50	0.628
InceptionV3	0.543
EfficientNetB0	0.649

Choosing a pre-trained CNN

Transfer Learning

If a model is trained on a large and general enough dataset, this model will effectively serve as a generic model of the visual world



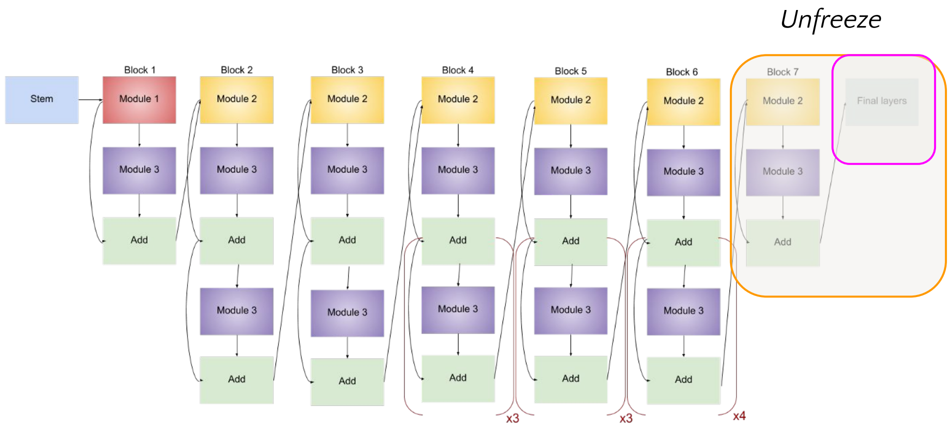


Model	Sample (Mean F1)
VGG16	0.588
ResNet50	0.628
InceptionV3	0.543
EfficientNetB0	0.649

Model	Train (Mean F1)	Test (Mean F1)
phash	0.596	0.613
ENetB0_TL	0.649	0.671

Fine Tuning

"fine-tune" the higher-order feature representations in the base model in order to make them more relevant for the specific task



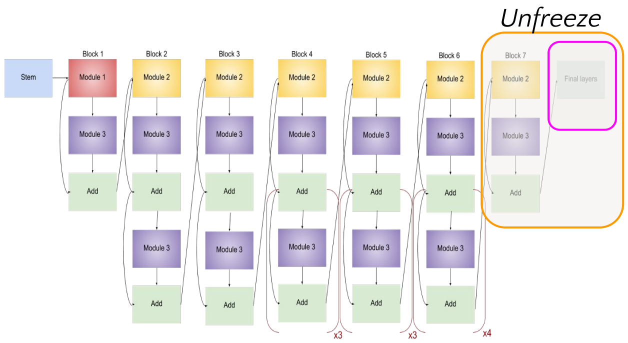
Unfreeze 1 layer

- Train 3 epochs
- Train 6 epochs

Unfreeze 1 module (several layers)

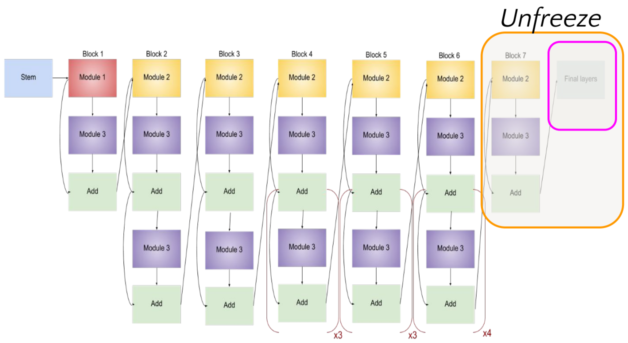
- Train 3 epochs
- Train 6 epochs
- Train 9 epochs

Fine Tuning



Model	Epochs	Train (Mean F1)	Test (Mean F1)
phash	-	0.596	0.613
ENetB0_TL	-	0.649	0.671
ENetB0_FT (1 Layer)	3	0.664	0.686
ENetB0_FT (1 Layer)	6	0.664	0.688
ENetB0_FT (1 Module)	3	0.681	0.696
ENetB0_FT (1 Module)	6	0.686	0.701
ENetB0_FT (1 Module)	9	0.686	0.701

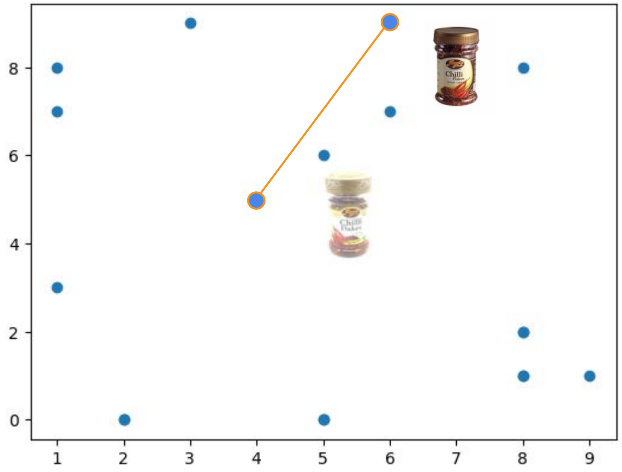
Fine Tuning



Model	Epochs	Train (Mean F1)	Test (Mean F1)
phash	-	0.596	0.613
ENetB0_TL	-	0.649	0.671
ENetB0_FT (1 Layer)	3	0.664	0.686
ENetB0_FT (1 Layer)	6	0.664	0.688
ENetB0_FT (1 Module)	3	0.681	0.696
ENetB0_FT (1 Module)	**6**	**0.686**	**0.701**
ENetB0_FT (1 Module)	9	0.686	0.701

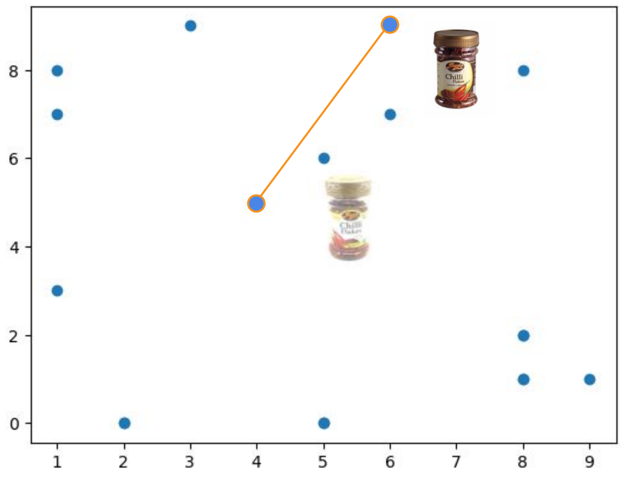
Alternative Metrics

Euclidean Distance

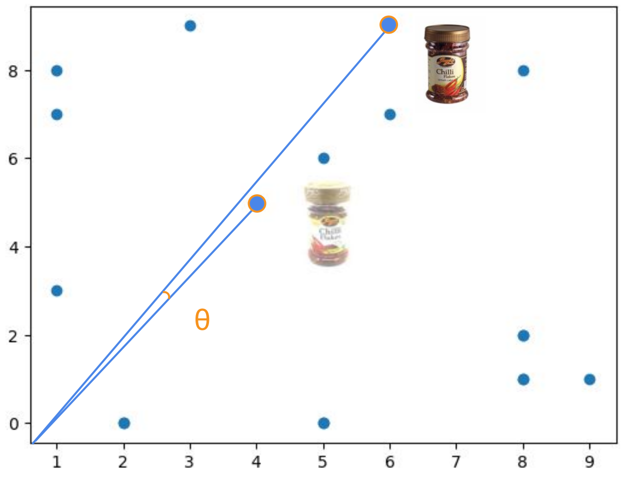


Alternative Metrics

Euclidean Distance

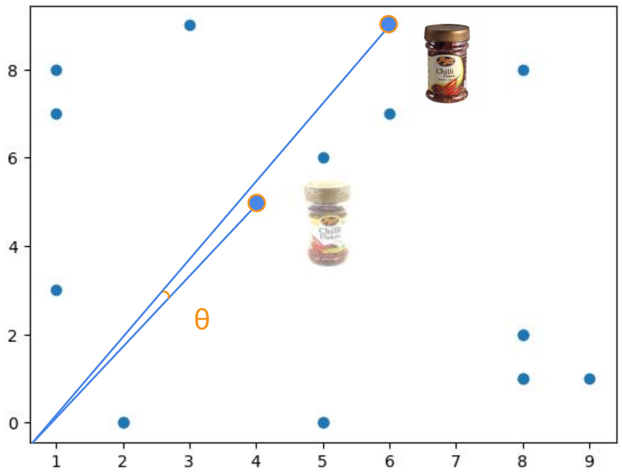


Cosine Distance



Alternative Metric

Cosine Distance



Model	Epochs/ (Metric)	Train (Mean F1)	Test (Mean F1)
phash	-	0.596	0.613
ENetB0_TL	-	0.649	0.671
ENetB0_FT (1 Layer)	6 (eucli)	0.664	0.688
ENetB0_FT (1 Module)	6 (eucli)	0.686	0.701
ENetB0_FT (1 Module)	6 (**cosine**)	**0.716**	**0.724**

Error Analysis

Error Analysis

Product	y_true	y_pred

Error Analysis

Product

y_true

y_pred

Error Analysis

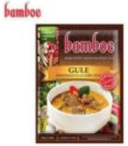
Product	y_true	y_pred
	 	

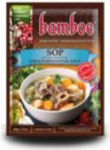
Error Analysis

Product

y_true

y_pred





Error Analysis

Product

y_true





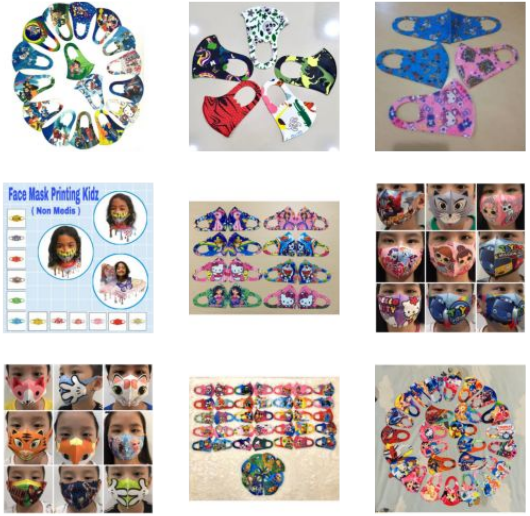



y_pred

Error Analysis

Product

y_true



y_pred

Error Analysis Notes

Model can only identify products with similar shape and form

False Positives

- May not negatively affect customer experience severely


False Negatives

- Model is unable to account for semantics of product
  Missing out on important recommendations


Conclusion and Recommendations

Conclusion

The model does well in predicting products that belong to the same category with a mean F1 score of 0.716 on the train data and 0.724 on the test data

Model can be improved by including other features which capture the product semantics

1. Product Title
2. Product Categories

Thank You!