Team project. Wroclaw University of Science and Technology. Created: 12.04.24r

 Link to github with project: https://github.com/c100rczyk/Team_project_PWR

New metric:

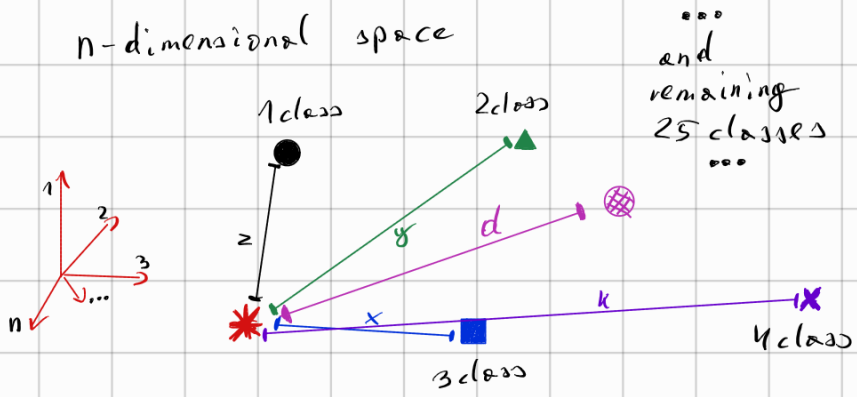
We have dataset with 30 images. Each image belongs to one class. So we have 30 classes - 30 different products.

Dataset: in our shop

image 1	image 2	image 3	image 4	...	image 30
apple A	apple B	tomato	Mango		Plum

From our convolutional model leaves feature vector that is different for each image. Different classes has visibly different vectors. So are in different part on n-dimensional space.

So next, we get distance from each class.
Distances on image: x,y,z,k,...

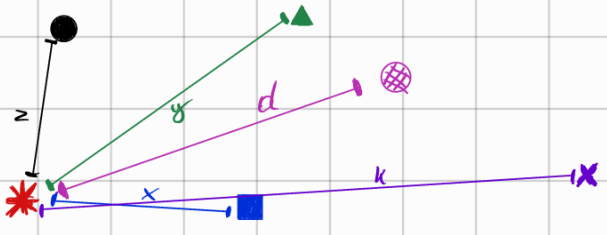


* photo of the scanned product.

When we have vector for scanned product and vecotrs for each of 30 classes in our shop database, we can calculate exact distances x,y,x,k... . We use this in next steps.

Each place has own number, by which we mean a penalty.

First place has 1.0 , so there is no penalty.
Second place has 0.98
Third place has 0.96
4-th has 0.94 and 5-th has 0.92.



Why such number? Because we accept for the purposes of this task 30 classes. The futher the real class is in the results, the worse the model performs. We want to determine the quality of this model in percantages, where 100% is the maximum, 50% is the result of random selection, and 0% is when the model places each correct class in the last 30-th place.

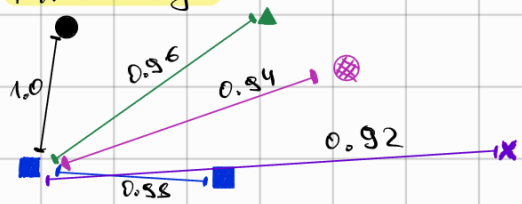
$$\frac{50}{30 \text{ classes}} \approx 1{,}67 \approx \underline{\underline{2\% \text{ step}}} \Rightarrow 1{,}0\ (100\%),\ 0{,}98\ (98\%),\ 0{,}96\ (96\%),\ \ldots$$

The 6-th and next places can have penalty numbers like: -10 or more.
If image is not in screen we definitely dont want such model. And we dont want that such situation occurs.

To get satisfaction results we need thousands of testing images. Then get value of each result and sum and divide by number of test samples. :

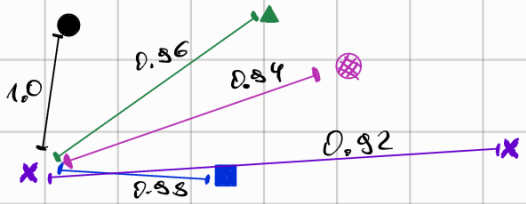
Example: 4 test images:

First image

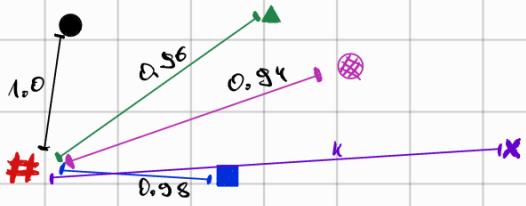


Second image

third image



4-th image



What we have:
First image is scanned and placed in space. Model says that the closest image is ●
Model says also that second closest image is ■ . This is correct one. So model is not as good as could be in this case.
So we get 0.98 to this prediction.

Second image is scanned and placed in space. Model says that the closest image is ●
This is correct one. So model is as good as could be in this case.
So we get 1.0 to this prediction.

...

Fourth image is scanned and placed in space. Model says that the closest image is
Model says also that second closest image is . And so on and so for.
Model dont give good result in 5 most important (closest) cases.
So we get -10 to this prediction.

Next we calculate: $\frac{0.98 + 1.0 + 0.92 + (-10)}{4} = -1,775$ ∴ -177%

If model would give us better results for each image: $\frac{0.98 + 1.0 + 0.92 + 0.96}{4} = 96,5\%$

In big test datasets e.g thousands of images , if model dont see the product, one, twice or even 3 Times it is still good enough.
Example:
12000 test images e.g (12000 times when client give his product)
Model dont saw the product in 3 cases.
So it is 1 product in 4000 products.