

# CLIP Augmentation for Image Search

Ingus Janis Pretkalnins<sup>1</sup>, Arturs Sprogis<sup>1</sup> and Guntis Barzdins<sup>2</sup><sup>a</sup>

<sup>1</sup>*Institute of Mathematics and Computer Science, University of Latvia, Raina blvd. 29, LV-1459, Riga, Latvia*

<sup>2</sup>*LETA, Marijas st. 2, LV-1050, Riga, Latvia*

Keywords: CLIP, Image Search, Probability as Extended Logic.

Abstract: We devised a probabilistic method for adding face recognition to the neural network model CLIP. The method was tested by creating a prototype and matching 1000 images to their descriptions. The method improved the text to image Recall @ 1 metric from 14.0% matches for CLIP alone to 21.8% for CLIP + method, for a sample size of 1000 images and descriptions.

## 1 INTRODUCTION

For some professions, like journalism and art design, finding appropriate images for their work is a near essential part of their work. Thus in the era of all types of databases growing ever larger, including image databases, the necessity of image search is growing ever larger.

Thankfully technology is not ignorant to this. Much research has gone into this topic.

Many search engines (e.g. Google, Bing, DuckDuckGo, etc.) provide an option for image search. Furthermore recent technology advances are opening the possibility of higher abstraction image feature extraction. By utilizing these advances, in this paper we hope to alleviate at least some of the image search problems troubling journalists.

We had a database of  $\sim 1.3$  million images from news reports. Most images in the database also had short Latvian descriptions of what is in the image. We were tasked with creating some way to easily search these images. More precisely, the search solution was to be designed so that journalists could easily find appropriate images for their news articles. For the purpose of scientific novelty, in this paper the descriptions are not used in the process of retrieving images (however they are used in the business solution). Instead, once trained, it only observes the contents of novel images.

As a base for the solution, we chose the neural network model CLIP (Radford et al., 2021). CLIP is a neural network that was trained to match images and

descriptions based on how likely they would be to appear together. It was conjectured that CLIP's abilities could quite straight forward be transferred to our task of finding appropriate images to a text search query. CLIP however has some short-comings. For instance it is known to recognize people badly (CLI, ) However for our task, the presence of a certain person in an image is very important.


This paper describes a probabilistic method for augmenting CLIP with facial recognition. The method utilizes the probability theory as extended logic (Jaynes, 2003). The same probabilistic model can likely be used for further augmentation. For example—logo recognition, building recognition, and other specific things that were unlikely to be in CLIP's training dataset.

In experiments it was determined that the approach does indeed improve performance. In a prototype augmenting CLIP with face recognition increased the percentage of correct text to image matches between 1000 images and descriptions from 14.0% to 21.8%.

## 2 RELATED WORK

There are many neural networks that match descriptions and images in a similar fashion as CLIP (Tiwary, 2021; Pham et al., 2021; Jia et al., 2021).

There has been work on integrating visual features with tag like features (Romberg et al., 2012; Kennedy and Naaman, 2008). However to the author's knowledge there isn't any research into specifically augmenting CLIP-like models, which is the main contri-

<sup>a</sup> <https://orcid.org/0000-0002-3804-2498>

bution of this paper.

The paper uses probability theory as an extension of logic. The methods of derivation used in the paper are described in the first 4 chapters of *Probability Theory: The Logic of Science* (Jaynes, 2003).

To integrate CLIP with face recognition, an extension of Platt scaling (Platt et al., 1999), temperature scaling (Guo et al., 2017), was used.

### 3 METHODOLOGY

This section contains more detailed descriptions of the probabilistic method for augmenting CLIP with face recognition. The section also contains descriptions of the parts used in building the prototype.

A rough sketch of the architecture of the prototype can be seen in Fig. 1.

#### 3.1 CLIP

OpenAI's CLIP is a transformer neural network model, that creates encoding vectors of 512 numbers for images and descriptions. One can then obtain a similarity value between an image and description, by normalizing the encoding vectors and computing their dot product.

This approach for computing similarity is quite fast. The image encodings only have to be computed once for each image in the database. After that, searching the database requires only computing the encoding of the text query, and calculating the dot product with all normalized image encodings. (In our case taking the dot product of 1.3 million vectors is feasible)

Throughout the paper we used the CLIP ViT-B/32 model pre-trained for English.

#### 3.2 Translation

Since the CLIP model was trained for English, but all of our data was in Latvian, we found it necessary to add a step for translating text queries from Latvian to English. The translation model used was EasyNMT (Tang et al., 2020) version "m2m\_100\_1.2B". While there were marginally better translation models available, EasyNMT was used, as it was easier to integrate into the solution.

#### 3.3 Model Derivation

In this section we'll derive a way of calculating the probability of each image corresponding to a certain text query. The method assumes a black box has

given us information about what people were recognized in the images and what people's names were recognized in the text query.

The model was created using *probability theory as extended logic*.

##### 3.3.1 Notation

$\neg A$ —the proposition that  $A$  is false;

$AB$ —the proposition that  $A$  and  $B$  are both true;

$A + B$ —the proposition that at least one of  $A$  and  $B$  are true;

$P(A|B)$ —the probability of  $A$ , given  $B$  is true;

$\prod_{i \in K} (A_i)$ —the proposition that all  $A_i$  are true;

$\sum_{i \in K} (A_i)$ —the proposition that at least one  $A_i$  is true.

The order of operations is  $\neg A, AB, A + B$ .

##### 3.3.2 Problem Definition

It was assumed that the face recognition in images and name recognition in text works like a black box. That is to say - the models simply give us boolean answers about which people's faces were detected in images, and which people's names were detected in the text query.

- We are given a description.
- We are given  $n$  images. It's assumed exactly one corresponds to the text query.
- The face and name recognizer, can recognize  $m$  different people.
- We are given information about what people the name recognizer thinks appear in the text query.

Let  $D_j$  be the proposition that person  $j$ 's name is recognized in the text query by the name recognizer.

Let  $D'_j$  be our information about person  $j$ 's recognition in the text query. In other words  $D'_j$  is  $D_j$ , if person  $j$  was recognized in the text query, and  $D'_j$  is  $\neg D_j$  if not.

- We know which faces are recognized in each image.

Let  $I_{ij}$  be the proposition that person  $j$  is recognized in image number  $i$ .

$I'_{ij}$  is defined analogously to  $D'_j$ .

- Let  $C_i$  be the proposition that the text query corresponds to the  $i$ -th image.

Exactly one  $C_i$  is true.

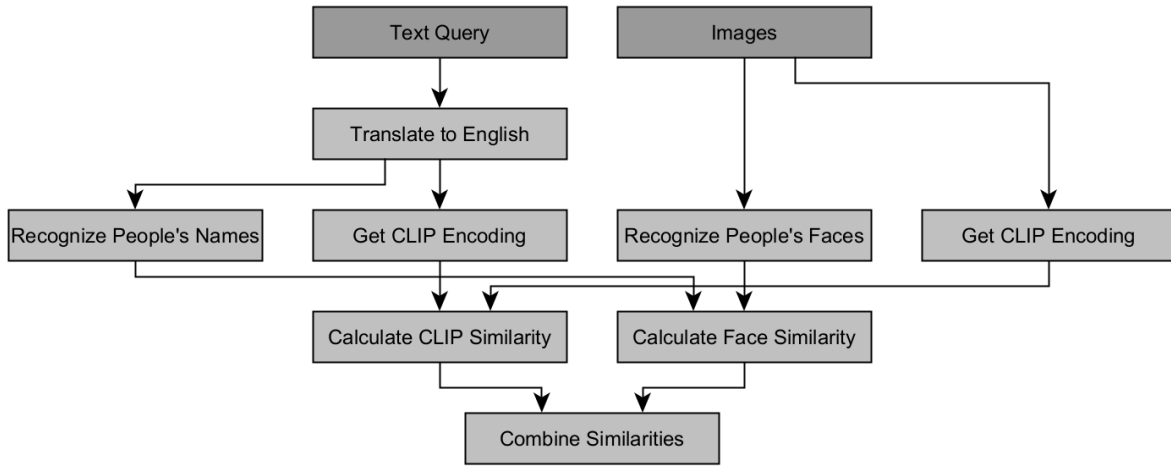


Figure 1: Pipeline with arrows showing how information flows between steps.

- Let  $M$  be all information from face and name recognition.  
i.e.

$$M = \prod_{i=1}^n \left( \prod_{j=1}^m (I'_{ij}) \right) \prod_{i=1}^m (D'_i) \quad (1)$$

- Let  $X$  be all of the information described above as well as other assumptions made during the derivation:  
 $X$  contains assumptions about independence  
 $X$  contains information to deduce  $P(D'_m|X)$  and  $P(D'_{ki}|D'_iC_kX)$ .  
 $X$  contains training data.  
 Prior probability distribution of recognizing people in images and descriptions.

Now the question that interests us is—how to calculate the probability  $P(C_i|MX)$ ? That is to say—what is the probability of the  $i$ -th image corresponding to the description, given all of our information?

### 3.3.3 Solution

Let's express the probability in a different form.

$$P(C_i|MX) = \frac{P(C_i|X)P(M|C_iX)}{P(M|X)} =$$

(We assumed one image corresponds to the text query, i.e. hypothesis exclusivity)

$$= \frac{P(C_i|X)P(M|C_iX)}{P(M|\sum_{j=1}^n (C_j)X)} =$$

$$= \frac{P(C_i|X)P(M|C_iX)}{P(M|\sum_{j=1}^n (C_j)|X)/P(\sum_{j=1}^n (C_j)|X)} =$$

(We assumed an image corresponds to the query)

$$= \frac{P(C_i|X)P(M|C_iX)}{P(M|\sum_{j=1}^n (C_j)|X)/1} =$$

(Use distributivity of logical or and hypothesis exclusivity)

$$= \frac{P(C_i|X)P(M|C_iX)}{\sum_{k=1}^n (P(MC_k|X))}$$

$$= \frac{P(C_i|X)P(M|C_iX)}{\sum_{k=1}^n (P(C_k|X)P(M|C_kX))} =$$

( $X$  doesn't contain information to discern between  $C_k$  thus the  $P(C|X)$ s cancel out)

$$= \frac{P(M|C_iX)}{\sum_{k=1}^n P(M|C_kX)}$$

We've reduced the problem to calculating all  $P(M|C_kX)$ .

Let's split up our information  $M$  into two parts  $M = M_k M_{-k}$ .

$$M_k = \prod_{i=1}^m (I'_{ki} D'_i) \quad (2)$$

In other words  $M_k$  is the information about faces recognized in the  $k$ -th image and names recognized in the text query.

$$M_{-k} = \prod_{i \neq k} \left( \prod_{j=1}^m (I'_{ij}) \right) \quad (3)$$

In other words  $M_{-k}$  will contain information about

faces recognized in all other images.

$$\begin{aligned} P(M|C_kX) &= \\ &= P(M_k M_{-k} | C_k X) = \\ &= P(M_k | M_{-k} C_k X) P(M_{-k} | C_k X) = \end{aligned}$$

(The fact that image  $k$  corresponds to the text query likely doesn't influence the probability of recognizing faces in unrelated images. So, we can simplify)

$$= P(M_k | M_{-k} C_k X) P(M_{-k} | X) =$$

(Recognizing faces in images likely doesn't significantly change the probability of recognizing faces and names in an unrelated image or text query)

$$= P(M_k | C_k X) P(M_{-k} | X)$$

Now let's separately find each of these probabilities. At first let's find  $P(M_{-k} | X)$ .

Let's make the assumption that finding concrete faces in an image, doesn't affect the probability of finding other faces in the image. More formally:

$$P\left(I'_{ij} \middle| \prod_{(k,l) \in A} (I'_{kl} | X)\right) = P(I'_{ij} | X), \quad (4)$$

where  $A \subseteq \{1, 2, \dots, n\} \times \{1, 2, \dots, m\} \setminus (i, j)$

This is probably one of the strongest assumptions made in the derivation. A counter example to this assumption could be—if You had a minister recognized in an image, it's much more likely that other politicians might also be in the image, at the same time it's less likely that, say, famous artists could appear in the same image. However one could hope, this isn't a huge problem for most images.

Using this assumption and the product rule repeatedly, we can obtain:

$$P(M_{-k} | X) = \prod_{i \in \{1, \dots, n\} \setminus \{k\}} \left( \prod_{j=1}^m P(I'_{ij} | X) \right) \quad (5)$$

To calculate  $P(I_{ij} | X)$  and  $P(\neg I_{ij} | X)$  we need to make some more assumptions.

We assume that the prior probabilities (before accounting for training data) are uniformly distributed and independent.

We also have to assume the probability is equal in all images.

After that  $P(I_{ij} | X)$  can simply be estimated as the amount of positive+1 divided by all cases+2.

Now let's look at  $P(M_k | C_k X)$ .

Repeatedly applying the product rule we can get:

$$P(M_k | C_k X) = \prod_{i=1}^m P(D'_i I'_{ki} | \prod_{j=i+1}^m (D'_j I'_{kj}) C_k X) \quad (6)$$

Now we do another strong analogous argument as previously. We assume recognizing people's faces in the image or names in the text query doesn't affect the probability of recognizing other people's faces in the image or names in the text query. More formally:

$$P\left(D'_i I'_{ki} \middle| \prod_{i \in A} (D'_i I'_{ki}) C_k X\right) = P(D'_i I'_{ki} | C_k X), \quad (7)$$

where  $A \subseteq \{1, 2, \dots, m\} \setminus \{i\}$

Using this assumption we can simplify  $P(M_k | C_k X)$  to:

$$P(M_k | C_k X) = \prod_{i=1}^m P(D'_i I'_{ki} | C_k X) \quad (8)$$

Now what's left is to find a way to calculate  $P(D'_i I'_{ki} | C_k X)$ .

$$\begin{aligned} P(D'_i I'_{ki} | C_k X) &= P(D'_i | C_k X) P(I'_{ki} | D'_i C_k X) = \\ &\text{(The } k\text{th image being correct doesn't} \\ &\text{affect the probability of finding} \\ &\text{something in the description.)} \\ &= P(D'_i | X) P(I'_{ki} | D'_i C_k X) \end{aligned} \quad (9)$$

Analogously to  $P(I_{ij} | X)$ ,  $P(D'_m | X)$  can be estimated by looking at how frequently names are found in the training data.

$P(I'_{ki} | D'_i C_k X)$  consists of 4 possibilities  $I_{ki} D_i$ ,  $\neg I_{ki} D_i$ ,  $I_{ki} \neg D_i$  un  $\neg I_{ki} \neg D_i$ . All of them can be estimated by looking at what gets recognized in images and their corresponding descriptions.

Putting this all together we can calculate  $P(C_k | MX)$ , which is what we set out to do.

Now a further problem arises—time complexity. Doing all the calculations naively would require  $O(n^2 \cdot m^2)$  time. Also there's a risk of floating point operation underflows.

### 3.3.4 Generality of Mathematical Model

We can notice that in this model  $D'_i$  and  $I'_{ki}$  doesn't have to necessarily represent names an faces of people. We could have just as well used them to represent brand logos, buildings, or similar objects that CLIP is unlikely to have seen in its training data.

## 3.4 Model Time Complexity Improvements

This section describes how to improve the time complexity of the probability calculation method to  $O(n \cdot (m_{\text{img}} + m_{\text{desc}}))$ , where  $m_{\text{img}}$  is the average amount of faces recognized in images across the database, and

$m_{\text{desc}}$  is the amount of names recognized in the text query.

$P(D_i I_{ki} | C_k X)$  can be calculated in constant time, i.e.  $O(1)$ .

$P(M_k | C_k X) = \prod_{i=1}^m P(D_i I_{ki} | C_k X)$  can be calculated in  $O(m)$  time. However let's notice that nearly all  $D_i$  and  $I_{ij}$  will be false. (We're exceedingly unlikely to find an image with 10'000 faces, or a description or text query with 10'000 names). So instead of multiplying together all  $P(D_i I_{ki} | C_k X)$ , we could pre-compute  $\prod_{i=1}^m P(-D_i - I_{ki} | C_k X)$ , and adjust for the cases where  $D_i$  or  $I_{ki}$  are true, by multiplying the result by  $\frac{P(D_i I_{ki} | C_k X)}{P(-D_i - I_{ki} | C_k X)}$ . The amount of differences will be no more than the amount of names recognized in the text query plus the amount of faces recognized in the image. That means the time complexity of computing all  $P(M_k | C_k X)$  would be  $O(n \cdot (m_{\text{desc}} + m_{\text{img}}))$ .

$P(M_{-k} | C_k X)$  can be computed analogously. That is to say, assuming  $I_{ij}$  is false and then correcting the cases where  $I_{ij}$  is true. That gives us a time complexity of  $O(n \cdot m_{\text{img}})$  for a single  $k$ . Which means computing it for all  $k$  would take  $O(n^2 \cdot m_{\text{img}})$ .

This however can be improved by noticing that

$$P(M_{-k} | C_k X) = \prod_{i \in \{1, \dots, n\} \setminus \{k\}} \left( \prod_{j=1}^m P(I_{ij} | X) \right) \quad (10)$$

differs on average in only about  $2 \cdot m_{\text{img}}$  multiplicands. So instead of recomputing everything, we can again just correct the differences in  $O(m_{\text{img}})$  time. This brings the time complexity down to  $O(n \cdot m_{\text{img}})$ . [Correct the fact that first time we have to calculate for all  $m$ ]

When computing

$$P(C_i | MX) = \frac{1}{n} \cdot \frac{P(M | C_i X)}{\sum_{j=1}^n P(M | C_j X)}, \quad (11)$$

we can of course reuse the divisor. This brings down the time for computing all  $P(C_i | MX)$  to  $O(n)$  given all  $P(M | C_j X)$ .

The total time complexity of the algorithm comes out to be  $O(n \cdot (m_{\text{img}} + m_{\text{desc}}))$ .

### 3.5 Float Errors

In order to avoid floating point overflows or underflows, one can take the logarithm of all probabilities during computation. Then use addition and subtraction instead of multiplication and division. Then the result only needs to be exponentiated back at the last step when calculating  $P(C_i | MX)$ , when summation is needed.

In order to avoid underflows in this last step, one can instead of calculating

$$\frac{P(M | C_i X)}{\sum_{j=1}^n P(M | C_j X)} \quad (12)$$

calculate

$$\frac{e^{\ln(P(M | C_i X)) - c}}{\sum_{j=1}^n e^{\ln(P(M | C_j X)) - c}}, \quad (13)$$

where  $c = \max_j (\ln(P(M | C_j X)))$

That is to say, before exponentiation normalize the logarithms, so the biggest exponential comes out to 1. This makes it so that only tiny probabilities get an underflow, which are insignificant anyway.

### 3.6 CLIP Augmentation

Using the probabilistic method described earlier we can calculate  $P(C_i | M_f X_f)$  ( $M$ , and  $X$  have been renamed to  $M_f$  and  $X_f$  for clarity).

After rescaling CLIP's similarity values can be interpreted as the inputs to a softmax that returns probabilities. That means we can interpret CLIP's similarity values as  $b \ln(P(C_i | M_c X_c)) + a$ . Where  $X_c$  is the knowledge the CLIP model has about the distribution of image-description pairs, and  $M_c$  is the image and description.

We can find  $b$  by using Platt scaling (Platt et al., 1999). Now in order to actually combine face recognition and CLIP we need to find  $P(C_i | M_f M_c X_f X_c)$ . We'll have to do some model independence assumptions in order to do this. We'll denote these assumptions  $X$ . So we need to find  $P(C_i | M_f M_c X_f X_c X)$ .

$$\begin{aligned} P(C_i | M_f M_c X_f X_c X) &= \\ &= \frac{P(C_i | M_c X_f X_c X) P(M_f | C_i M_c X_f X_c X)}{P(M_f | M_c X_f X_c X)} = \\ &= \frac{P(C_i | M_c X_f X_c X) P(M_f | C_i M_c X_f X_c X)}{\sum_{i=1}^n P(C_i | M_c X_f X_c X) P(M_f | C_i M_c X_f X_c X)} = \end{aligned}$$

(We assume the models are independent under the correct hypothesis  $C_i$ )

$$= \frac{P(C_i | M_c X_f X_c X) P(M_f | C_i X_f X)}{\sum_{i=1}^n P(C_i | M_c X_f X_c X) P(M_f | C_i X_f X)}$$

(Assume the face model is doesn't give any extra information about the distribution of images and descriptions.)

$$= \frac{P(C_i | M_c X_c X) P(M_f | C_i X_f X)}{\sum_{i=1}^n P(C_i | M_c X_c X) P(M_f | C_i X_f X)}$$

(Model independence assumptions shouldn't affect the probability estimates of a single models)

$$\begin{aligned}
 &= \frac{P(C_i|M_cX_c)P(M_f|C_iX_f)}{\sum_{i=1}^n P(C_i|M_cX_c)P(M_f|C_iX_f)} = \\
 &\text{(Take the logarithm and exponentiate)} \\
 &= \frac{e^{(\ln(P(C_i|M_cX_c)) + \ln(P(M_f|C_iX_f)))}}{\sum_{i=1}^n e^{(\ln(P(C_i|M_cX_c)) + \ln(P(M_f|C_iX_f)))}} = \\
 &\text{(Multiply both sides by a constant)} \\
 &= \frac{e^{(\ln(P(C_i|M_cX_c)) + a) + \ln(P(M_f|C_iX_f))}}{\sum_{i=1}^n e^{(\ln(P(C_i|M_cX_c)) + a) + \ln(P(M_f|C_iX_f))}} \quad (14)
 \end{aligned}$$

Thus, if we know  $\ln(P(C_i|M_cX_c)) + a$  (The constant  $a$  doesn't really matter to our calculation) and  $\ln(P(M_f|C_iX_f))$ , the calculation of probability is quite straight forward.

This method of augmenting can be quite straight forward generalized assuming independence of all models given the correct hypothesis, and assuming independence of CLIP and all models. Assuming we have  $q$  models the result simply comes out to:

$$\begin{aligned}
 P(C_i|M_cX_c \prod_{j=1}^q (M_jX_j)X) &= \\
 &= \frac{e^{(\ln(P(C_i|M_cX_c)) + a) + \sum_{j=1}^q (\ln(P(M_j|C_iX_j)))}}{\sum_{i=1}^n e^{(\ln(P(C_i|M_cX_c)) + a) + \sum_{j=1}^q (\ln(P(M_j|C_iX_j)))}} \quad (15)
 \end{aligned}$$

In simpler terms, one can just combine the models in a naive-Bayes sort of style, assuming all the model independence assumptions are good enough approximations.

### 3.7 Prototype

In order to test whether the probabilistic model works, we created a prototype that recognizes people's faces in images, and people's names in descriptions/text queries. We then used the model to augment CLIP by the method described above.

To recognize names in text, we used a pre-trained neural network for named entity recognition from the NLP library Flair (Akbik et al., 2019). To recognize faces in images, we used the python *face\_recognition* (King, 2009; Geitgey, 2020) that can locate and recognize faces.

To associate names with faces, we found images where one face was recognized, and the image description contained one recognized name. Let's call these images face-shots. The recognition doesn't happen perfectly. However that really doesn't matter, since the probabilistic model doesn't assume face or

name recognition happens perfectly. Worse recognition reduces the amount of information the probabilistic model is able to derive from the recognition, but it doesn't break it.

To understand which face-shots actually correspond to the same person, the face-shots were then clustered. Two face-shots were placed in the same cluster, if their *face\_recognition* distance was less than 0.5 and their name Character Error Rate (an edit distance based metric) was less than 0.2. A threshold of error was allowed for names because of different possible conjugations or translations of Latvian names.

In order to detect whether a face belonged to a known person, the closest cluster which had a face with no more than 0.5 face distance was chosen. To detect a person's name, the cluster with the least CER was chosen, as long as the CER was no more than 0.2.

The probabilities for  $P(I_{ki}'|D_i' C_k X)$  were estimated by looking at ~55 thousand images and their descriptions from 2019.

The prototype was tested on different images from 2020.

The results of augmenting CLIP with this prototype can be seen in table 1.

## 4 DATASET AND EVALUATION

### 4.1 Data

We had access to a database of about 1.3 million images and their Latvian descriptions from news reports. The database was provided to us by the Latvian news agency LETA.

### 4.2 Recall @ k Metric

We used the Recall @ k metric, to compare how well images get assigned to descriptions.

To calculate it, we take a sample of images and their descriptions. Then use the model to assign a probability of the image corresponding to the description, for each description-image combination. Recall @ k is the percentage of descriptions, for which the correct image was among the top rated k images.

It's assumed that higher performance in these metrics should correlate well with performance in our task, which is searching images from a database by typing a text query.

The performance on these metrics doesn't scale linearly with the sample size. Thus when doing comparisons, the same sample size must be chosen.

Table 1: Results of tests with and without augmentation using the probabilistic model described.

Metric	R@1	R@5
CLIP	0.140	0.306
Faces	0.069	0.120
CLIP+Faces	<b>0.218</b>	<b>0.396</b>

## 5 EXPERIMENTS

### 5.1 Efficacy of Prototype

A test was run to see whether augmenting CLIP with face recognition according to the described method improved results.

Two tests were run on a sample of the same 1000 randomly sampled images and their descriptions from 2020.

In one test only CLIP was used. In the other CLIP was augmented using the proposed probabilistic model. Face-shot clusters and probability estimates were obtained from images and descriptions from 2019. Results can be seen in table 1.

## 6 CONCLUSION AND FUTURE WORK

We got 21.8% R@1 metric by using the face recognition prototype as compared to 14.0% for CLIP alone, which shows that the described method of CLIP augmentation does work.

The method of CLIP augmentation can likely be used for different types of objects aswell, like company logos, buildings or other things that are unlikely to have been in the CLIP training data.

However there is much room for future work. The boolean model of names and faces being detected likely discards some useful information. If the model could be modified to take into account information about distance to clusters, that could be a potential avenue for improvement.

Another possibility for future work might be to relax some of the independence assumptions. For instance one could cluster people who are more likely to appear together, and use that to improve the estimate of whether an image is relevant.

## 7 NOTE ON RESPONSIBLE USE

The authors used face recognition only for purposes of enhancing image search of public figures in LETA's

internal image database for journalists. The authors strongly advise against it's use in cases where it could undermine people's privacy.

## ACKNOWLEDGEMENTS

The research was supported by ERDF project 1.1.1.1/18/A/045 at IMCS, University of Latvia.

This research is funded by the Latvian Council of Science, project No. lzp-2021/1-0479.

## REFERENCES

- Clip face recognition. <https://openai.com/blog/clip/>. [Online; accessed 16-November-2021].
- Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.
- Geitgey, A. (2020). Python face\_recognition library. <https://pypi.org/project/face-recognition/>. [Online; accessed 16-November-2021].
- Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks.
- Jaynes, E. T. (2003). *Probability Theory: The Logic of Science*.
- Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q. V., Sung, Y., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision.
- Kennedy, L. and Naaman, M. (2008). Generating diverse and representative image search results for landmarks. In *in Pro c. 17th Int. Conf. World Wide Web*, pages 297–306.
- King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758.
- Pham, H., Dai, Z., Ghiasi, G., Liu, H., Yu, A. W., Luong, M.-T., Tan, M., and Le, Q. V. (2021). Combined scaling for zero-shot transfer learning.
- Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.
- Romberg, S., Lienhart, R., and Hörster, E. (2012). Multimodal image retrieval. *International Journal of Multimedia Information Retrieval*, 1.

- Tang, Y., Tran, C., Li, X., Chen, P., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. (2020). Multilingual translation with extensible multilingual pretraining and fine-tuning. *CoRR*, abs/2008.00401.
- Tiwary, S. (2021). Turing bletchley: A universal image language representation model by microsoft. [https://www.microsoft.com/en-us/research/blog/turing-bletchley-a-universal-image-language-representation-model-by-microsoft/?OCID=msr\\_blog\\_Bletchley\\_hero](https://www.microsoft.com/en-us/research/blog/turing-bletchley-a-universal-image-language-representation-model-by-microsoft/?OCID=msr_blog_Bletchley_hero). [Online; accessed 20-December-2021].

