

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Компьютерных сетей и систем

Кафедра Информатики

РЕФЕРАТ

по курсу «Машинное обучение»

Задача распознавания лиц

Студент:
гр. 758641
Ярош Г.И.

Проверил:
Заливако С. С.

Минск, 2019

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
Сферы применения	3
КЛАССИЧЕСКИЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ ЛИЦ	4
1. Eigenfaces	4
Описание алгоритма	4
2. Fisherfaces	5
Описание алгоритма	6
РАСПОЗНАВАНИЕ ЛИЦ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ	7
1. FaceNet	7
Сиамские сети	7
Триплет потерь	8
Архитектура сети	9
2. VGG-Face	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ВВЕДЕНИЕ

Задача распознавания лиц является одной из наиболее приоритетных и популярных задач в сфере машинного обучения и компьютерного зрения. Каждый день мы встречаемся с системами уличного видеонаблюдения, системы распознавания лиц применяются в офисах вместо ключей и магнитных карт, мобильные телефоны используют системы распознавания лиц для разблокировки.

Современные системы распознавания лиц могут даже определять является ли рассматриваемое лицо реальным, что предотвращает попытки взлома, основанные на использовании фотографий и других изображений человека.

К задаче распознавания лиц иногда относят следующие смежные задачи:

- Обнаружение лица на изображении. Обычно данная задача предшествует самому распознаванию лица. Результатом ее обычно является область или области изображения, где находятся лица людей, которые можно впоследствии использовать для распознавания.
- Верификация лица. Является более узкой проблемой, и ставит задачу определить сходство входного изображения лица и некоторого проверочного. Распознавание лиц в общем случае ставит задачу по входному изображению определить входит ли оно в некоторую группу известных людей.

Сферы применения

Распознавание лиц находит применения во многих сферах жизни. Вот только некоторые из них:

- Распознавание лиц используется при выдаче документов, удостоверяющих личность, в сочетании с другими биометрическими технологиями.
- Идентификация лиц используется при проверке на таможне для сопоставления лица человека с изображением в биометрическом паспорте.
- Распознавание лиц широко применяется в сфере обеспечения правопорядка для поиска правонарушителей, используя записи уличных камер наблюдения.
- Распознавание лиц используется в различных системах авторизации, разблокировке мобильного телефона, получение доступа в офис.

КЛАССИЧЕСКИЕ АЛГОРИТМЫ РАСПОЗНАВАНИЯ ЛИЦ

1. *Eigenfaces*

Eigenfaces относится к подходам к решению задачи распознавания лиц, который основан на подсчете дисперсии коллекции изображений лиц и использует эту информацию для кодирования и сравнения изображений отдельных лиц в целом (в отличие от основанного на деталях или основанного на особенностях метода). В основе Eigenfaces лежит метод главных компонент, где главные компоненты распределения пикселей изображений лиц называются собственными лицами или, другими словами, собственными векторами ковариационной матрицы множества изображений лиц, где изображение N пикселей считается точкой (или вектором) в N мерном пространстве. Идея использования метода главных компонент для представления человеческих лиц была разработана Сировичем и Кирби (Sirovich and Kirby 1987) и использовалась Турком и Пентлендом (Turk and Pentland 1991) для обнаружения и распознавания лиц. Подход Eigenface, как полагают многие, является первой рабочей технологией распознавания лиц, и он послужил основой для одного из лучших коммерческих продуктов технологии распознавания лиц. Со времени его первоначальной разработки и публикации было создано много расширений оригинального метода и много новых разработок в системах автоматического распознавания лиц. Eigenfaces все еще часто рассматриваются как метод для сравнения по базовой линии, чтобы продемонстрировать минимальную ожидаемую эффективность такого рода систем.

Описание алгоритма

Для корректной работы алгоритма изображения лица нормализуются для выравнивания глаз и рта, а затем все повторно дискретизируются с одинаковым разрешением в пикселях. Собственные лица затем извлекаются из данных изображения с помощью анализа главных компонент (РСА) следующим образом:

1. Пусть дано N изображений лиц, каждое из которых представлено вектором размером d :

$$X = \{x_1, x_2, \dots, x_N\}$$

2. Для всех векторов подсчитываем среднее значение μ :

$$\mu = \frac{1}{N} \sum_{i=1}^N x_i$$

3. Вычисляем ковариационную матрицу C :

$$C = \frac{1}{N} \sum_{i=1}^N (x_i - \mu)(x_i - \mu)^T$$

4. Вычисляем собственные значения λ_i и собственные векторы v_i ковариационной матрицы C :

$$Cv_i = \lambda_i v_i, \forall i = 1, 2, \dots, N$$

5. Выбираем k собственных векторов, которые соответствуют k наибольшим собственным значениям.
6. Вычисление k главных компонент вектора X_i происходит по формуле:

$$y_i = V^T(x_i - \mu),$$

$$\text{где } V = (v_1, v_2, \dots, v_k)$$

Распознавание изображения с помощью Eigenface происходит следующим образом:

1. Для всех N тренировочных изображений x_i посчитываются собственные лица y_i :

$$y_i = V^T(x_i - \mu),$$

$$\forall i = 1, 2, \dots, N$$

2. Для распознаваемого изображения x лица также подсчитываются собственные лица y :

$$y = V^T(x - \mu)$$

3. В качестве критерия похожести распознаваемого изображения на изображение в тренировочной выборке можно использовать евклидово расстояние:

$$d_i = \|y - y_i\|^2$$

Если расстояние меньше изначально заданного порога, то можно рассматривать, что входное изображение x распознано, как x_i .

Подход Eigenface был важным шагом на пути к распознаванию лиц в компьютерном зрении. Тем не менее, метод чувствителен к изменению освещения, масштаба, позы, выражения лица и окклюзии. Чтобы алгоритм работал хорошо, лицо должно быть представлено в виде спереди, в соответствующем масштабе, в аналогичном освещении, в определенном (обычно нейтральном) выражении и не включенным. Для решения множества проблем, с которыми сталкиваются реальные системы распознавания лиц, было предложено множество модификаций и расширений с момента первоначальной реализации метода.

2. Fisherfaces

Метод главных компонент (PCA), который является основой метода Eigenfaces, находит линейную комбинацию функций, которая максимизирует общую дисперсию данных. Хотя это, несомненно, мощный способ представления данных, он не учитывает какие-либо классы, и поэтому при отбрасывании компонент может быть потеряно много дискриминационной информации. Алгоритм Fisherface использует метод, называемый линейным дискриминантным Анализом. Метод выполняет уменьшение размерности с учетом класса объекта и был изобретен знаменитым математиком Р. А. Фишером. Он успешно использовал его для классификации цветов в своей работе 1936 года «Использование нескольких измерений в таксономических задачах». Чтобы найти комбинацию признаков,

которая лучше всего разделяет классы, линейный дискриминантный анализ максимизирует соотношение между классами и разброс внутри классов, а не максимизирует общий разброс. Идея проста: одни и те же классы должны близко располагаться друг к другу, в то время как разные классы находятся как можно дальше друг от друга в пространстве более низкой размерности.

Описание алгоритма

Пусть дано M классов изображений $X_j = (x_1, x_2, \dots, x_{n_j})$, $\forall j = 1, 2, \dots, M$. Алгоритм Fisherface заключается в следующем:

1. Различия внутри класса могут быть оценены с помощью матрицы рассеяния, вычисляемой следующим способом:

$$S_w = \sum_{j=1}^M \sum_{i=1}^{n_j} (x_{ij} - \mu_j)(x_{ij} - \mu_j)^T,$$

где μ_j – среднее значение для класса j

2. Похожим образом вычисляются межклассовые различия используя межклассовую матрицу рассеяния:

$$S_b = \sum_{j=1}^M (\mu_j - \mu)(\mu_j - \mu)^T,$$

где μ – среднее значение для всех классов

3. Далее находятся такие базисные векторы $V = (v_1, v_2, \dots, v_k)$, что значение S_w минимизируется, а значение S_b максимизируется. Решение может быть найдено с помощью разложения по собственным значениям:

$$S_b V = S_w V \Lambda,$$

где Λ - матрица собственных значений, V - матрица собственных векторов.

4. Найденные собственные векторы, соответствующие ненулевым значениям называются фишер-лица.

Распознавание изображения с помощью Fisherfaces происходит следующим образом:

4. Для всех N тренировочных изображений x_i посчитываются фишер-лица y_i :

$$y_i = V^T(x_i - \mu),$$

$$\forall i = 1, 2, \dots, N$$

5. Для распознаваемого изображения x лица также подсчитываются подсчитывается фишер-лица y :

$$y = V^T(x - \mu)$$

6. В качестве критерия схожести распознаваемого изображения на изображение в тренировочной выборке можно использовать евклидово расстояние:

$$d_i = \|y - y_i\|^2$$

Если расстояние меньше изначально заданного порога, то можно рассматривать, что входное изображение x распознано, как x_i .

РАСПОЗНАВАНИЕ ЛИЦ С ПОМОЩЬЮ НЕЙРОННЫХ СЕТЕЙ

В настоящее время активным образом развиваются технологии нейросетевого моделирования. Многие задачи, которые раньше представляли сложность для классических алгоритмов, теперь решаются с высокой эффективностью.

Задача распознавания лиц получила новые эффективные решения благодаря появлению и развитию глубоких сверточных нейронных сетей, а также постоянно увеличивающейся вычислительной эффективности современных компьютеров.

В данной части предлагается рассмотреть подходы, применяемые при построении и обучении нейронных сетей для решения задач распознавания лиц.

1. FaceNet

Одной из фундаментальных архитектур нейронных сетей, для распознавания лиц стала FaceNet. Она была опубликована группой исследователей компании Google в 2015 году. В ней были применены новые приемы обучения, такие как триплет потеря, сиамские сети. Данная архитектура проложила путь для создания следующего поколения систем распознавания лиц с использованием машинного обучения.

Сиамские сети

FaceNet использует специфический тип архитектуры, получившей название сиамской сети (рис. 3).

Сиамские сети состоят из двух идентичных нейронных сетей, каждая из которых имеет одинаковые точные веса. Во-первых, каждая сеть принимает одно из двух входных изображений в качестве входных данных. Затем выходы последних слоев каждой сети отправляются в функцию, которая определяет, степень подобия изображений

В процессе обучения, алгоритм оптимизации применяется не к каждому выходу сети, а к общей функции потерь, принцип работы которой описан в следующем пункте.

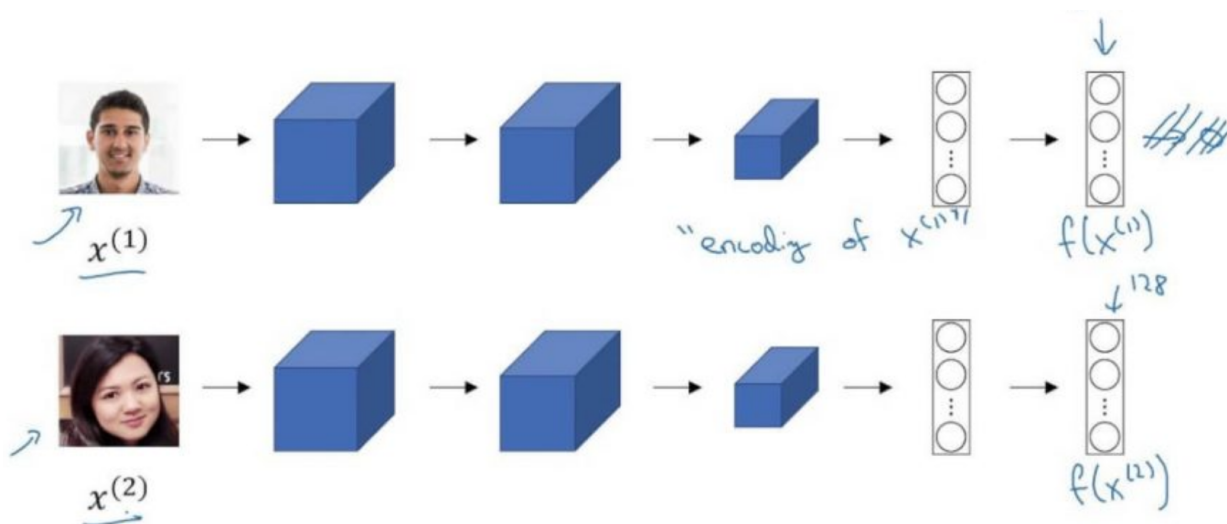


Рис. 3. Принцип работы сиамских сетей.

Триплет потерь

В качестве функции потерь, FaceNet использует специфическую функцию, которой поставлена цель минимизировать степень различия между лицами, принадлежащими к одному классу и максимизировать различия между лицами одного класса.

Данная функция использует следующий принцип формирования обучающей выборки. В выборке выделяются тройки изображений, одно из которых является тестовым (якорь), другое относится к тому же классу, а третье относится к другому (рис. 4). При чем два последних изображения подаются на вход нейронной сети, и соответственно функция потерь вычисляет степень похожести «верного» и неверного изображения на якорь.

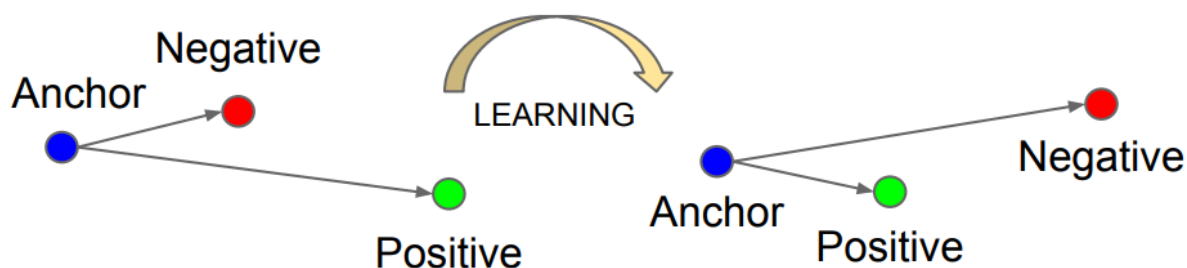


Рис. 4. Принцип формирования троек изображений.

Также имеет смысл выбирать тройки таким образом, чтобы верное и неверное изображения были максимально похожи.

Сама функция потерь для такого типа обучения сети выглядит следующим образом:

$$Loss = \sum_{i=1}^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+,$$

где $f(x_i^a)$ — значение энкодинга якоря,

$f(x_i^p)$ – значение энкодинга верного изображения,

$f(x_i^n)$ – значение энкодинга неверного изображения.

Коэффициент α – константа обеспечивающая, что сеть не будет оптимизировать функцию потерь следующим способом $f(x_i^a) - f(x_i^p) = f(x_i^a) - f(x_i^n) = 0$.

При выборе троек изображений следует стремиться к следующему соотношению: при фиксированном x_i^a , необходимо найти такие x_i^p и x_i^n , что:

$$\begin{aligned} \|f(x_i^a) - f(x_i^p)\|_2^2 &\rightarrow \max \\ \|f(x_i^a) - f(x_i^n)\|_2^2 &\rightarrow \min \end{aligned}$$

Выбор троек изображений в процессе обучения сети может происходить следующим образом:

- Выбирать тройки изображений каждые n шагов обучения. Для подсчета сходства можно использовать последнее состояние обучаемой сети.
- Выбирать тройки в процессе итерации из текущего минибатча изображений. При использовании данного подхода следует формировать батчи большого размера.

Правильный выбор троек изображений для обучения критичен для быстрой сходимости.

Архитектура сети

В описанной выше работе рассматривалось несколько архитектур глубоких сверточных сетей для распознавания лиц. Пример базовой архитектуры приведен на рисунке 5.

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

Рис. 5. Пример арихтектур сети, рассматривающей в работе.

Другие архитектуры сетей, приведенные в работе состоят из Inception слоев и имеют меньшее число параметров. Сравнение результатов тестирования приведено на рисунке 6.

architecture	VAL
NN1 (Zeiler&Fergus 220×220)	87.9% ± 1.9
NN2 (Inception 224×224)	89.4% ± 1.6
NN3 (Inception 160×160)	88.3% ± 1.7
NN4 (Inception 96×96)	82.0% ± 2.3
NNS1 (mini Inception 165×165)	82.4% ± 2.4
NNS2 (tiny Inception 140×116)	51.9% ± 2.9

Рис. 6. Сравнение точностей моделей, рассматриваемых в работе.

2. VGG-Face

Другой архитектурой глубокой сверточной нейронной сети, использующейся для распознавания лиц является VGG-Face. Она была разработана группой исследователей в университете Оксфорда.

VGG-Face использует более глубокую архитектуру, чем остальные популярные сети для распознавания лиц (рис. 7).

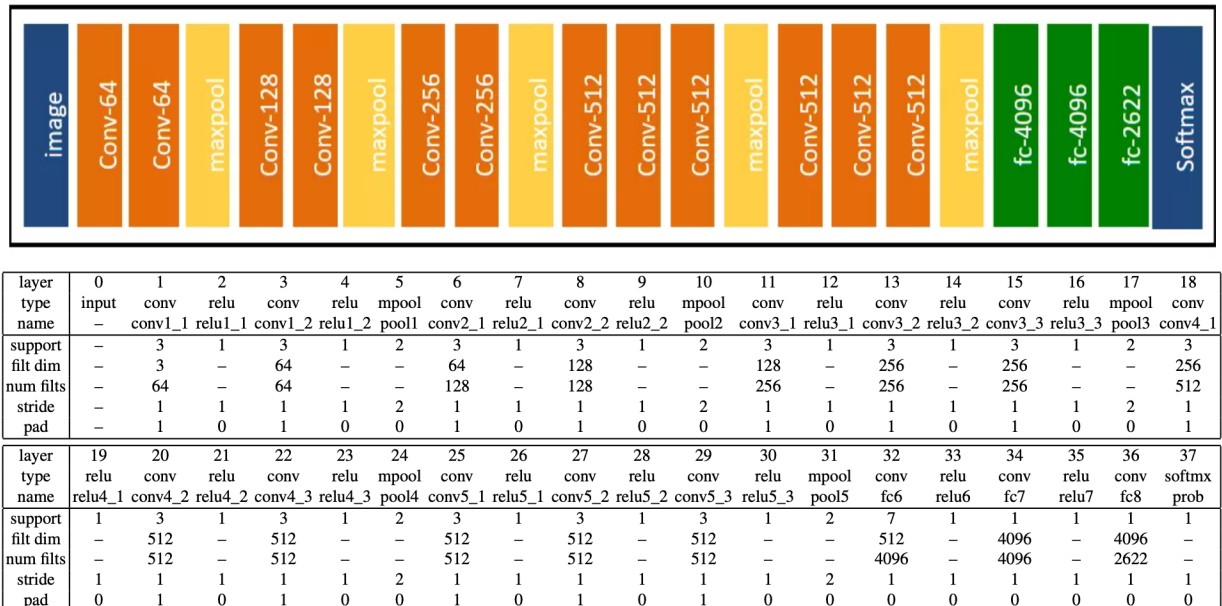


Рис. 7. Архитектура VGG-Face

VGG-Face также использует триплет потерь при обучении. Выбор троек изображений осуществляется на каждой итерации из батча. Сравнение VGG-Face с другими популярными архитектурами приведены на рисунках 8, 9. В сравнении использовались наборы данных YouTube Faces In the Wild и Labeled Faces In the Wild.

No.	Method	# Training Images	# Networks	100%-EER	Accuracy
1	Video Fisher Vector Faces	-	-	87.7	83.8
2	DeepFace (Facebook)	4 M	1	91.4	91.4
4	DeepID-2,2+,3		200	-	93.2
5	FaceNet (Google)	200 M	1	-	95.1
7	Ours (VGG Face)	2.6 M	1	97.60	97.4

Рис. 8. Сравнение эффективности на наборе данных YouTube Faces In the Wild.

No.	Method	# Training Images	# Networks	Accuracy
1	Fisher Vector Faces	-	-	93.10
2	DeepFace (Facebook)	4 M	3	97.35
3	DeepFace Fusion (Facebook)	500 M	5	98.37
4	DeepID-2,3	Full	200	99.47
5	FaceNet (Google)	200 M	1	98.87
6	FaceNet+ Alignment (Google)	200 M	1	99.63
7	Ours (VGG Face)	2.6 M	1	98.78

Рис. 9. Сравнение эффективности на наборе данных Labeled Faces In the Wild.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Eigenfaces [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.scholarpedia.org/article/Eigenfaces>.

[2] Fisherfaces [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.scholarpedia.org/article/Fisherfaces>

[3] FaceNet: A Unified Embedding for Face Recognition and Clustering [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://arxiv.org/pdf/1503.03832.pdf>.

[4] Deep Face Recognition [Электронный ресурс]. – Электронные данные. – - Режим доступа: <http://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf>.

[5] Deep Face Recognition Poster [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/poster.pdf>.