Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет «Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

КУРСОВАЯ РАБОТА

на тему

Исследование функций похожести в задаче оценки позы человека с использованием порождающих моделей

Выполнила студентка группы 142, 2 курса,
Вальтер Дарья Дмитриевна

Научный руководитель:
к. ф.-м. н., доцент,
Конушин Антон Сергеевич

Консультант:
Шахуро Владислав Игоревич

Москва 2016

Содержание

1 Аннотация

С развитием вероятностного моделирования порождающие (генеративные) модели становятся все более перспективны в решении задач компьютерного зрения таких, как комплексное описание распознаваемой сцены. Одним из важных компонентов при решении этой задачи является эффективный выбор функции, задающей степень похожести наблюдаемого и сгенерированного изображения.

В данной работе рассматривается способы задания функции похожести с использованием различных признаковых описаний изображений на примере задачи распознавания позы человека. В качестве функций похожести используются вручную заданные метрики такие, как сравнение карт краев изображений, а также l2-расстояние в пространстве нейросетевых признаков с разных слоев нейросети. Эти метрики также сравниваются с обученной с помощью нейросетей функцией похожести.
Экспериментальная оценка показала, что использование обученной функции похожести обеспечивает наиболее быструю сходимость модели.

With recent progress on probabilistic modeling, top-down generative models have been becoming more perspective in the computer vision task. "Analysis by synthesis"frameworks can describe complex scenes more richly and flexibly. Similarity function between scene hypotheses and observed image is one of the important components of generative models.
This work describes several different similarity functions. It compares the performance of contours maps to l2-distance between neural net features and to the learnt similarity function by neural network.
Experiments show that learnt similarity function provides the fastest and most robust model's performance.

2 Введение

2.1 Методы компьютерного зрения

Комплексное описание сцены на изображении и распознавание объектов является одной из основных задач компьютерного зрения. Самыми распространенными методами для решения этой задачи являются методы машинного обучения, в первую очередь дискриминативные (bottom-up) методы, такие как глубинные нейронные сети или леса решающих деревьев.

С помощью дискриминативных методов успешно решаются задачи классификации изображений: например, с помощью сверточной сети AlexNet в [1]. Однако минусами дискриминативного подхода можно считать:

- Потребность в обучающих выборках большого размера: трудоемкость сбора и разметки большой выборки
- Переобучение
- Медленная скорость обучения
- Черная коробка - сложность интерпретации

Нисходящие (top-down) или порождающие методы основаны на моделировании распознаваемой сцены. Сцена задается с помощью вероятностной модели, параметры которой являются случайными величинами из разных семейств распределений. Суть метода состоит в том, что выдвигаются гипотезы о значении параметров: расположении объектов, размере, освещении и т.д., затем гипотезы сравниваются с наблюдениями и последовательно улучшаются.

Достоинством порождающих методов является то, что их гораздо легче интерпретировать, чем дискриминативные методы. Генеративные модели строятся на основании физически осмысленных элементов. Кроме этого, в отличие от дискриминативных методов, порождающие модели позволяют оценить правдоподобие ответа-гипотезы при условии наблюдаемого изображения, а не точечную оценку, которые способны находить дискриминативные модели. В ряде исследований было показано, что генеративные модели могут анализировать сцены наиболее полным и гибким образом, достигая большой точности. Однако порождающие методы не настолько широко распространены из-за следующих особенностей:

- Необходимость моделирования специфичных сцен и сложность подобного моделирования
- Медленная скорость сходимости модели

Качество порождающей модели существенно зависит от метода сравнения гипотез и наблюдаемого изображения. В этой работе исследуются функции похожести гипотезы и наблюдаемого изображения в порождающей модели оценки позы человека. Проводилось сравнение функций похожести на основании того, как быстро модель сходится.

3 Постановка задачи

3.1 Задача распознавания позы человека.

Формальная постановка задачи заключается в следующем. На вход алгоритму подаются наблюдаемые изображения $I_1, \ldots, I_k$ с изображением позы человека.

Пусть $X_1, \ldots, X_N$ - множество меток классов объектов, соответствующих позам человека. В результате работы алгоритма каждому изображению присваивается метка $X_i \in X_1, \ldots, X_N$.

В случае генеративного подхода, описанного в **4.1**, на выход алгоритма подается вектор $(p_1, \ldots, p_n)$ - параметры модели, описание сцены-гипотезы. По известному вектору $(p_1, \ldots, p_n)$ можно восстановить метку класса.

3.2 Цель и задачи курсовой работы

Цель курсовой работы - изучение функций похожести изображений-гипотез и наблюдаемого изображения в задаче распознавания позы человека с помощью порождающей модели.

Для выполнения цели в рамках данной работы были поставлены следующие **задачи**:

- Провести обзор литературы о порождающем подходе.
- Провести обзор методов задания функции похожести изображений и выбрать наиболее перспективные для использования в порождающих моделях.
- Реализовать выбранные методы
- Провести экспериментальную оценку выбранных методов

4 Обзор существующих методов

4.1 Принцип порождающего подхода

В данной работе используется вероятностный язык для распознавания сцен Picture [2]. Общая схема порождающего подхода в распознавании сцен, описанная в [2], состоит из следующих компонент:

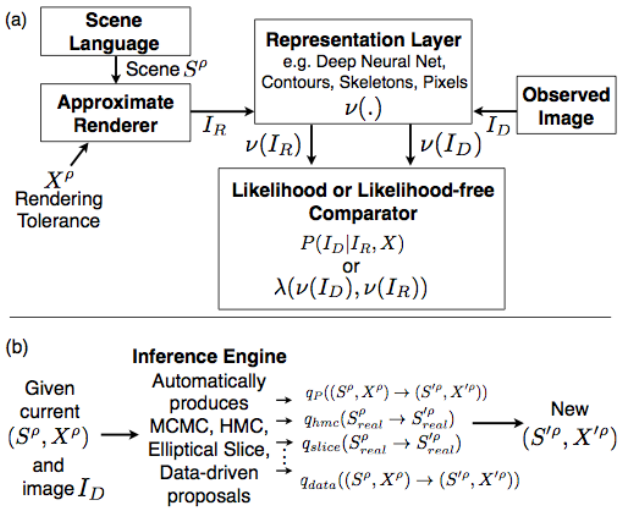


Рис. 1: Общая схема порождающего подхода в задаче распознавания

- **(a) Вход: наблюдаемое изображение.** На вход алгоритму подается наблюдаемое изображение I_D
- **Язык описания и движок рендеринга.** Вероятностная модель задается совокупностью случайных величин из различных семейств распределений, определяющих параметры объектов (геометрия, цвет, положение камеры, освещение). Сцена $S^r ho$ описывается совокупностью значений этих параметров. Графический движок рендерит изображение I_R по заданной сцене.
- **Уровень представления изображений.** Для компактного описания и сравнения изображения-гипотезы и наблюдаемого изображения используются различные признаковые описания. Функция $\nu(.)$ от изображения возвращает его признаковое описание. В данной работе будут описаны два вида признаков: карты краев изображения, нейросетевые признаки со сверточных и последних полносвязных слоев.
- **Функция сравнения представления изображений.** Для оценки похожести двух изображений по признаковым описаниям в данной работе используется вероятностная функция $Log - likelihood(I_D|I_R, \lambda(\nu(I_D), \nu(I_R)))$ - логарифм правдоподобия изображения-гипотезы, где $\lambda(\nu(I_D), \nu(I_R))$ - это функция похожести двух изображений.
- **(b) Генератор гипотез и движок вероятностного вывода.** Генератор гипотез и движок вероятностного вывода по имеющейся информации (наблюдаемое изображение, текущие параметры сцены, похожесть предыдущих гипотез на наблюдаемое изображение) генерирует новые параметры сцены, которые со временем делают синтетическое изображение все более похожим на наблюдаемое.
- **Выход: последовательность изображений-гипотез.** В результате работы алгоритма генерируется последовательность изображений, сходящаяся к изображению, наиболее похожему на наблюдаемое, исходя из значения Log-likelihood.

4.1.1 Генераторы гипотез

Сложность при генерации гипотез в порождающих моделях заключается в том, что вероятностное распределение модели представляет собой сложную смесь различных распределений, возможно и дискретных, и непрерывных. Тогда задача сэмплирования решается с помощью особых генераторов гипотез. Формально, генераторы гипотез - алгоритмы, которые получают на вход вероятностное распределение, заданное плотностью $p(x)$, и на выходе выдает сэмплы $x^1, ..., x^n$ из данного распределения. Основой идеей, на которой основаны генераторы гипотез является то, что, если равномерно выбирать точку под графиком функции плотности распределения p, то x-координата этой точки будет взята по распределению p (x-координата случайной точки будет встречаться с вероятностью, пропорциональной значению функции плотности в этой точке).

Сэмплирование с отклонением (rejection sampling).

Простейший метод сэмплирования - сэмплирование с отклонением.

Пусть задано другое распределение $q(x)$, из которого легко сэмплировать и которое приближает распределение $p(x)$, то есть $\exists c : \forall x cq(x) > p(x)$. Сэмплирование с отклонением работает следующим образом:

1. сэмплируется x из распределения $q(x)$
2. берётся случайное число u равномерно из интервала [0, cq(x)]
3. вычисляется $p(x)$
4. если $p(x) > u$, то x добавляется в сэмплы, а если меньше (т.е. если u не попало под график плотности $p(x)$), то x отклоняется.

Для того, чтобы этот метод работал, надо, чтобы распределение $cq(x)$ достаточно хорошо приближало $p(x)$. При больших размерностях $p(x)$ алгоритм становится неэффективным, т.к. большая часть кандидатов будет отвергаться. Кроме этого, алгоритм имеет проблемы с распределениями, сосредоточенными в узких областях.

Алгоритм Метрополиса-Гастингса.

Суть этого алгоритма основана на той же идее: точки берутся равномерно под графиком функции. Однако подход теперь другой: вместо того, чтобы пытаться приблизить распределение $p(x)$ другим распределением $q(x)$, алгоритм строит случайное блуждание под графиком функции, переходя от одной точки к другой, и время от времени берет текущую точку блуждания в качестве сэмпла.

Теперь $q = q(x|x^t)$ - распределение, которое зависит от последнего состояния x^t. Очередная точка, сэмплируемая из распределения q, принимается, если плотность p увеличивается, и отклоняется с некоторой вероятностью, если плотность p уменьшается. Очередная итерация алгоритма начинается с состояния x^t и заключается в следующем:

1. сэмплируется x^* из распределения $q(x|x^t)$
2. вычисляется $a = min(1, \frac{p(x^*)q(x^t|x^*)}{p(x^t)q(x^*|x^t)})$ (С поправкой на ассиметричность распределения q)
3. с вероятностью a: $x^{t+1} = x^*$, с вероятностью $1 - a$: $x^{t+1} = x^t$

4.2 Методы сравнения изображений

4.2.1 Карты краев. Детектор Canny

Этот метод основан на сопоставлении краев - контуров изображений. В исходной демке human pose, написанной на Picture, с которой мы будем работать, в качестве базового решения используется эта метрика. Для извлечения контуров из изображения используется детектор краев Canny.

Край - это точка резкого изменения значений функции интенсивности изображения. Большинство детекторов краев основаны на том, что берется градиент изображения и находятся локальные максимумы.

Детектор Canny, подробно описанный в [3], определяет контуры изображения в 4 шага:

1. Убрать шум и лишние детали из изображения
2. Рассчитать градиент изображения
3. Сделать края тонкими (edge thinning)
4. Связать края в контуры (edge linking)

Недостатками метода является чувствительность к линейным преобразованиям объекта, сдвигам и вращению объекта.

4.2.2 Гистограммы ориентированных градиентов. Дескриптор SIFT.

Дескриптор SIFT, основанный на локальных гистограммах ориентированных градиентов, был предложен в 1999 году в статье [4].

Алгоритм SIFT включает в себя следующие шаги:

1. На изображениях выделяются ключевые точки.
2. Для ключевых точек подсчитываются дескрипторы.
3. По совпадению дескрипторов выделяются соответствующие друг другу ключевые точки на двух изображениях.

Дескрипторы подсчитываются следующим образом. Окрестность ключевой точки делится на четыре квадратных сектора. В каждом пикселе внутри каждого сектора вычисляется градиент изображения, его направление и модуль. Затем модули градиентов умножаются на вес, экспоненциально убывающий с удалением от ключевой точки. Веса нужны для того, чтобы избежать резких изменений значения дескриптора при небольших изменениях положения окна, и для того, чтобы градиенты, удаленные от центра дескриптора, вносили меньший вклад в его значение.

По каждому сектору собирается гистограмма направлений градиентов, причем каждое вхождение взвешивается модулем градиента. Дескриптор SIFT представляет собой вектор, полученный из значений всех элементов гистограмм направлений, и состоит из 128 компонент. Дескриптор нормируется, чтобы повысить его устойчивость к изменениям яркости.

Достоинствами дескрипторов SIFT является инвариантность относительно масштабирования, изменения положения объекта, вращение объекта или камеры.

Существующие недостатки: чувствительность к условиям освещения, нечеткое выделение объектов относительно фона, плохое выделение объектов без четко выраженной текстуры.

4.2.3 Нейросетевые признаки

Сверточные нейронные сети активно применяются для решения различных задач компьютерного зрения: классификации и распознавания изображений [1], задачам регрессии [5] и др. Так в [5] была обучена нейронная сеть для распознавания позы и построения heatmap с координатами и пространственным расположением позы человека.

Однако, нейронные сети могут быть применимы и для задач сопоставления дескрипторов. В качестве дескрипторов изображения можно взять нейросетевые признаки с различных сверточных и полносвязных слоев уже обученной нейросети и рассматривать изображения в пространстве нейросетевых признаков.

В [6] в качестве функции похожести изображений бралось l2-расстояние между нейросетевыми признаками различных сверточных слоев. В качестве обученных нейросетей использовалась AlexNet и специально обученная нейросеть без заранее заданных меток класса по методу, описанному в [7]. Было показано, что дескрипторы нейросети, обученной с помощью обучения без учителя превосходят дескрипторы AlexNet в задаче сопоставления изображений. То есть, если AlexNet, обученная при помощи меток классов, подходит для решения задач классификации, то к задаче сопоставления дескрипторов применимы нейросети, обученные без учителя.

Но при этом было показано, что дескрипторы с обеих нейросетей значительно превосходят SIFT-дескрипторы применительно к поставленной задаче. Сопоставление нейросетевых дескрипторов достигает большей точности при различных преобразованиях изображений (линейных, нелинейных, изменении освещения, поворотах, масштабировании) по сравнению с SIFT-дескрипторами. Наилучшим выбором нейросетевых признаков при этом являются признаки с последнего сверточного слоя.

4.2.4 Методы обучения функции похожести с помощью нейросетей

Другой группой функций похожести являются функции, обученные с помощью нейросетей. На вход такой нейросети подаются два изображения, а значением целевой функцией является 1, если изображения полностью совпадают, и 0 в обратном случае.

Для обучения функции похожести в [8] описаны подходящие для этой задачи нейросетевые архитектуры.

- **Siamese.** Архитектура siamese сети состоит из двух модулей. Нижняя нейронная сеть состоит из двух ветвей, состоящих из одинаковых слоев (сверточных, слоев понижения размерности, ReLU) с одинаковыми весами. После этого выходы из обеих ветвей конкатенируются и подаются на вход верхней нейросети, состоящей из двух полносвязных и одного ReLU слоя. Таким образом, нижняя нейронная сеть по сути вычисляет нейросетевые дескрипторы каждого из изображений, а верхняя нейронная сеть представляет собой функцию похожести.

- **Псевдо-siamese.** Архитектура псевдо-siamese нейросети в точности такая же как и siamese, только веса двух ветвей обучаются независимо. Это позволяет увеличить количество настраиваемых параметров.

- **Двухканальная нейросеть.** В отличие от предыдущих архитектур двухканальная нейросеть не содержит двух ветвей - нет явного вычисления дескрипторов. Два изображения в градациях серого подаются на вход как одно двухканальное изображение.

В [8] было показано, что обученные метрики превосходят SIFT-дескрипторы по точности (метрика mAP, mean average precision) на тренировочных датасетах.

4.3 Заключение. Выбор метода

Проведенный обзор показал, что современные методы сопоставления изображений с использованием нейросетевых признаков и обученных функций похожести превосходят по точности ранее используемые дескрипторы SIFT.

Метрики, достигающие наилучших результатов в задаче сопоставления фрагментов изображения - это метрики, обученные с помощью двухканальных и siamese нейросетей. Трудность использования этих нейросетей заключается в том, что они требуют большие обучающие выборки и специальную процедуру обучения. В этой работе было решено использовать уже обученную нейронную сеть MatchNet [9], представляющую собой siamese архитектуру.

Обзор также показал, что использование нейросетевых признаков применительно к задаче сопоставления изображений было изучено в очень ограниченном числе работ. Однако нейросетевые признаки, обладающие большой информативностью и обобщающей способностью, являются хорошим дескриптором для описания изображения. Поэтому в данной работе было решено использовать дескрипторы VggNet [10], нейросети, показавшей в задаче распознавания изображений лучший результат, чем AlexNet.

Выбранные методы задания функции похожести решено было сравнить с исходным методом - сравнения карт краев, используемом в Picture.

5 Реализация

5.1 Используемые программные пакеты

5.1.1 Picture

Для решения задачи определения позы человека использовалась порождающая модель human pose, написанная на языке Picture. В работе использовалась модель human pose, портированная на язык программирования Python.

Опишем исходную порождающую модель human pose.

Вход. На вход программе подается RGB изображение I_D размерности $120 \times 160 \times 3$.

Вероятностное описание сцены. Сцена S^ρ описывается набором значений случайных величин, соответствующих источнику света, положению камеры, положению и поворотам частей человеческого тела. Формально, $S^\rho = [s_1, \ldots, s_n]$, где s_i - значение i-ого параметра модели. $p_1, \ldots, p_n$ - плотности соответствующих случайных величин.

Графический движок. В качестве графического движка используется Blender.

Генератор гипотез. В качестве алгоритма сэмплирования используется алгоритм Метрополиса-Гастингса.

Представление сгенерированных изображений . В качестве представления изображений $\nu(I)$ используются карты краев, полученные фильтром Canny.

Функция сравнения. В качестве функции сравнения используется вероятностная функция - логарифм функции правдоподобия $Log-likelihood$ сцены-гипотезы при условии наблюдаемого изображения. Функция правдоподобия задается следующим образом:

$$\mathcal{L} = p(S^\rho) p_\epsilon(\lambda(\nu(I_D), \nu(I_R)))$$

$$p(S^\rho) = \prod_{i=1}^{n} p_i(s_i)$$

где $\lambda(\nu(I_D), \nu(I_R))$ - функция похожести представлений наблюдаемого и сгенерированного изображений. p_ϵ - плотность распределения на ошибку модели. В исходной модели human pose P_ϵ - нормальное распределение со средним 0 и стандартным отклонением 0.35.

Логарифм правдоподобия берется для большей устойчивости значений.

5.1.2 Caffe

Для работы с нейросетями использовался фреймворк Caffe [11]. Библиотека Caffe написана на C++, имеет интерфейсы на языках Python и Matlab. При работе с нейросетями использовался Python-интерфейс Caffe.

С помощью Caffe с использованием нейросетей AlexNet и VggNet с уже обученными весами из изображений извлекались нейросетевые признаки.

Опишем архитектуру нейросети AlexNet. (Архитектура VggNet отличается от AlexNet незначительно). AlexNet содержит сверточные, субдискретизационные (max-pooling) и полносвязные (dense) слои.

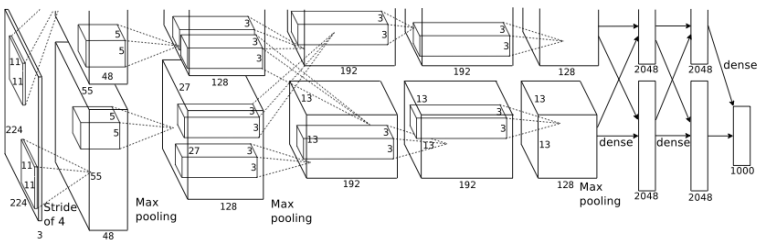
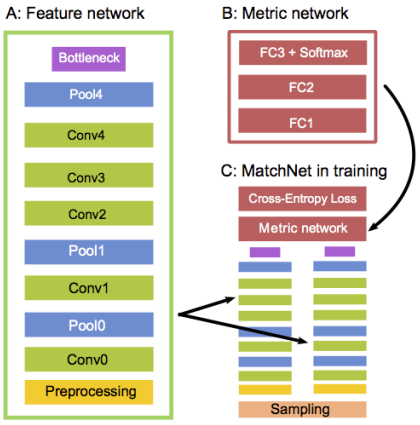


Рис. 2. Архитектура сети AlexNet

MatchNet. С помощью Caffe с использованием сети MatchNet [9] вычислялось значение обученной функции похожести для пар изображений. Описание архитектуры MatchNet:



Name	Type	Output Dim.
Conv0	C	$64 \times 64 \times 24$
Pool0	MP	$32 \times 32 \times 24$
Conv1	C	$32 \times 32 \times 64$
Pool1	MP	$16 \times 16 \times 64$
Conv2	C	$16 \times 16 \times 96$
Conv3	C	$16 \times 16 \times 96$
Conv4	C	$16 \times 16 \times 64$
Pool4	MP	$8 \times 8 \times 64$
Bottleneck	FC	-
FC1	FC	**1024**
FC2	FC	**1024**
FC3	FC	**1024**

Рис. 3. Архитектура сети MatchNet

5.2 Построение решения задачи

На первом этапе необходимо было реализовать функции сравнения представлений изображений.

l2-расстояние между нейросетевыми признаками.

1. Из изображения извлекаются нейросетевые признаки. Для этого изображение подается на вход обученной нейросети VggNet, выполняется прямой проход послойного вычисления значений нейронов в сети. В качестве признаков в экспериментальной оценке брались значения 5 сверточного слоя, 7 и 8 полносвязных слоев.

   Признаки с полносвязных слоев представляют собой массивы из n значений. (В 7 полносвязном слое - 4096 значений, в 8 полносвязном слое - 1000 значений).

   Признаки с 5 сверточного слоя представляют собой трехмерные массивы $512 \times 17 \times 17$ - 512 фильтров 17×17 пикселей каждый.

2. После того, как признаки извлечены для наблюдаемого и сгенерированного изображений, вычисляется l2-расстояние. Для полносвязных слоев:

$$l2(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

   Для сверточного слоя (размерности $A \times B \times B$) :

$$l2(x, y) = \sqrt{\sum_{i=1}^{A}\sum_{j=1}^{B}\sum_{k=1}^{B}(x_i jk - y_i jk)^2}$$

Обученная с помощью MatchNet функция похожести.

1. На вход нижней нейросети MatchNet, вычисляющей дескрипторы, подается изображение в градациях серого размерности 64×64. Следовательно, исходные изображения, наблюдаемое и гипотезу, необходимо привести к нужной размерности. Значения пикселей приводятся к значениям $\in [0, 1]$. Изображения подаются на вход по очереди.

2. Посчитанные признаки двух изображений подаются на вход верхней сети для подсчета похожести. На выходе получается значение $\in [0, 1]$ - вероятность совпадения.

На втором этапе необходимо было сравнить скорость сходимости порождающей модели при использовании выбранных функций похожести. В качестве скорости сходимости была взята скорость возрастания графика $Log - likelihood$ от числа итераций.

Для корректного сравнения графиков $Log - likelihood$ для различных функций похожести требовалось подобрать правильные параметры распределения ошибки модели P_ϵ. Этот подбор необходим, так как значения метрик разницы между картами краев, l2-расстоянием между нейросетевыми признаками и вероятность совпадения изображений, полученная на выходе MatchNet, отличаются на порядки. Так как исходное распределение $P_\epsilon \sim N(0, 0.35)$, то ошибка отклонения изображения-гипотезы от наблюдаемого изображения предполагается очень близкой к нулю для того, чтобы $Log - likelihood$ принимал относительно большие значения.

Для того, чтобы решить эту проблему, было решено эмпирически подобрать параметр P_ϵ для каждой функции похожести. Для этого было необходимо сгенерировать выборку из очень близких изображений с помощью Picture, измерить значения метрик на полученной выборке и найти выборочные дисперсии для каждой из метрик. Значения выборочных дисперсий подставить как параметр P_ϵ.

6 Экспериментальное тестирование

6.1 Проведение экспериментов

В качестве исходных данных использовалось изображение человека размерности $120 \times 160 \times 3$.

Рис. 4. Тестовое изображение

6.1.1 Эмпирический подбор дисперсий.

С помощью исходной демки human pose было сгенерировано 150 изображений, очень незначительно отличающихся друг от друга. После этого для каждой пары, составленной из этих изображений (22350 пар), были подсчитаны значения построенных функций похожести.

Таблица 1. Посчитанные значения стандартных отклонений:

Функция похожести	Стандартное отклонение
Карты краев	6.06
MatchNet	0.000001835
VggNet, полносвязный слой fc8	16.75
VggNet, полносвязный слой fc7	23.5
VggNet, сверточный слой conv5	183.89

6.1.2 Сравнение графиков *Log-likelihood* для разных функций похожести.

Для наблюдаемого изображения, подаваемого на вход, запускалась модель human pose. Количество итераций задавалось равным 1000. Модель запускалась несколько раз с каждой из функций похожести. Для воспроизводимости результатов и сравнения влияния метрик на сходимость, в программном коде задавалось значение k в numpy.random.seed(k), определяющее значения случайных величин в программе.

В среднем в ходе работы генератора гипотез (алгоритма сэмплирования Метрополиса-Гастингса) отвергалось порядка 70% гипотез. Для остальных гипотез вычислялось значение $Log - likelihood$.

Были подсчитаны значения $Log - likelihood$ по итерациям для каждой из функций похожести. Отдельно проводилось сравнение графиков $Log - likelihood$ при использовании нейросетевых признаков с различных слоев VggNet.

1. **Нейросетевые признаки с разных слоев.**

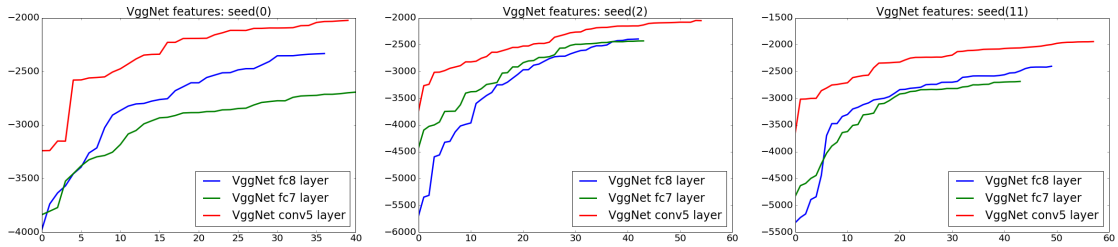


Рис. 5. Графики $Log - likelihood$ для l2-метрики.

Из графиков видно, что $Log - likelihood$ принимает наибольшие значения при использовании признаков conv5. Однако это может быть связано как с тем, что l2-расстояние в пространстве признаков conv5 принимают наиболее близкие значения, так и с неточностью в вычислении эмпирической дисперсии для параметра модели ошибки.

В целом, l2-метрика в пространстве нейросетевых признаков не показала хороших результатов в задаче сопоставления изображений ни с точки зрения похожести гипотез на наблюдаемое изображение, ни с точки зрения скорости сходимости модели.

Вероятно это связано с тем, что обученная сеть VggNet, решающая задачу классификации, относит позы человека к близким классам, и разница между нейросетевыми признаками разных поз незначительна.

При этом значения $Log - likelihood$ перестают расти в среднем после 50 итерации - по сравнению с другими функциями похожести сходимость медленная.

При одинаковых значениях k, ядра генерации случайных чисел, l2-метрика в сравнении с картой краев и MatchNet дает меньшую точность приближения к наблюдаемому изображению. Полученные изображения-гипотезы с последней итерации при разных запусках программы:

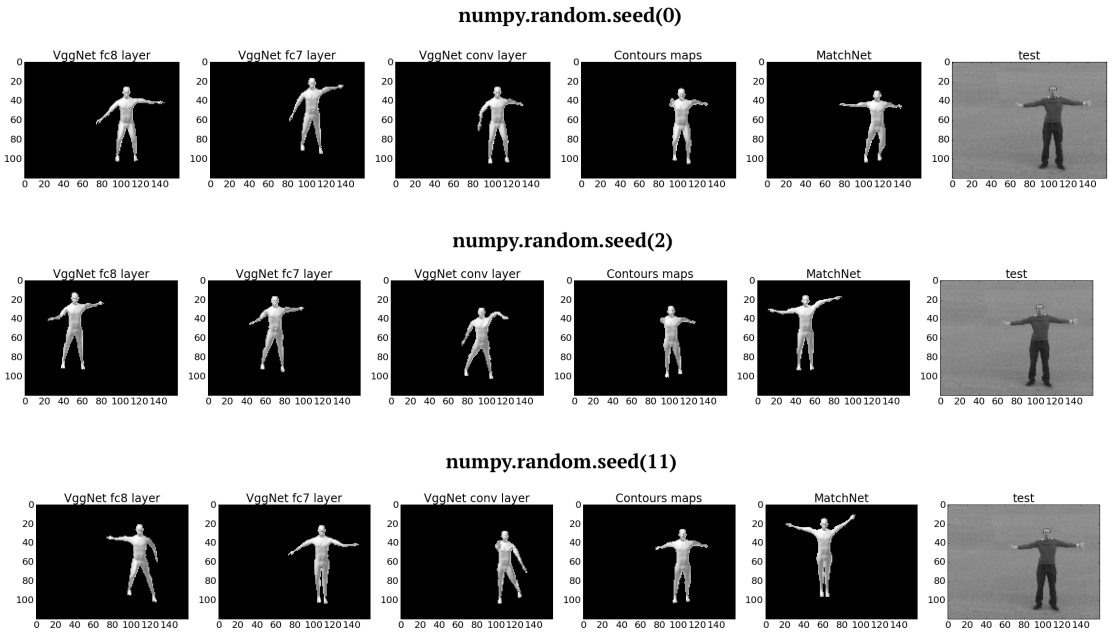


Рис. 6. Изображения, полученные с последних итераций для разных функций похожести.

2. **Метрика MatchNet.**

Функция похожести, обученная с помощью MatchNet, показала самую быструю скорость сходимости модели на всех запусках программы. В среднем модель сходится, начиная с 23 итерации.

Примеры, когда MatchNet превосходит карты краев по точности приближения к наблюдаемому изображению и по скорости сходимости модели:

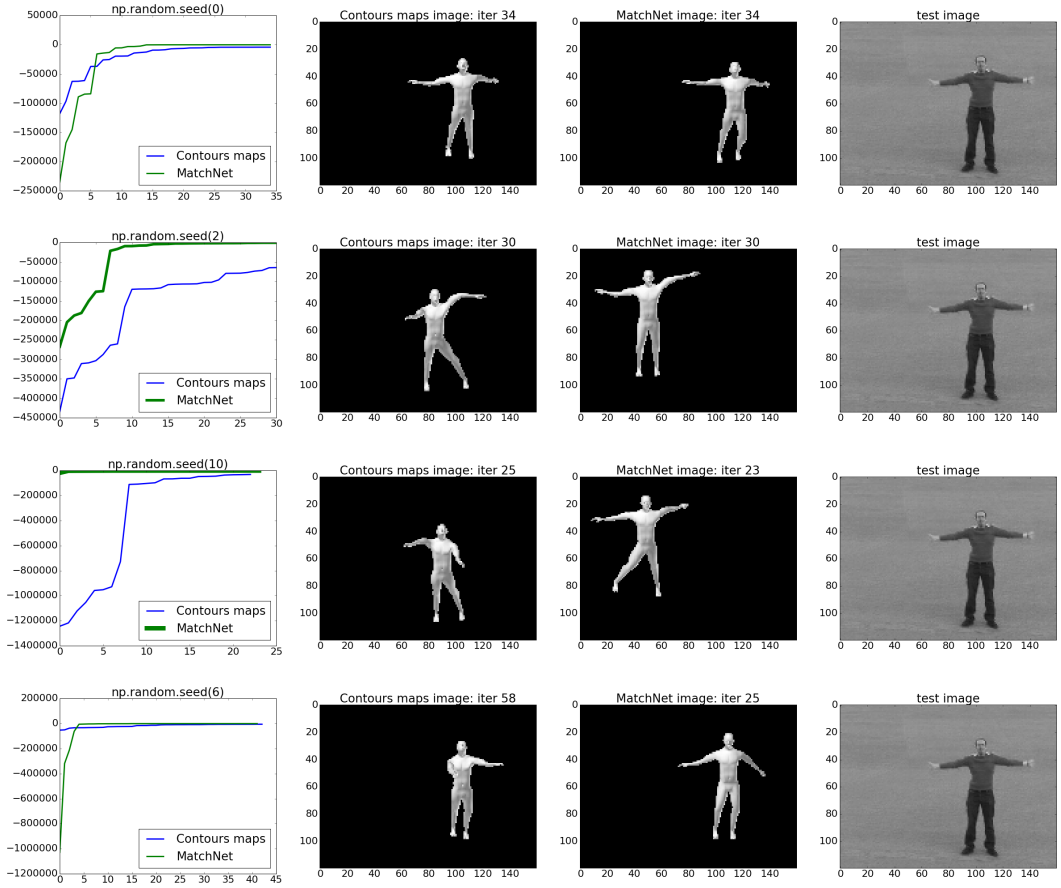


Рис. 7. MatchNet превосходит карты краев по точности приближения и по скорости сходимости модели

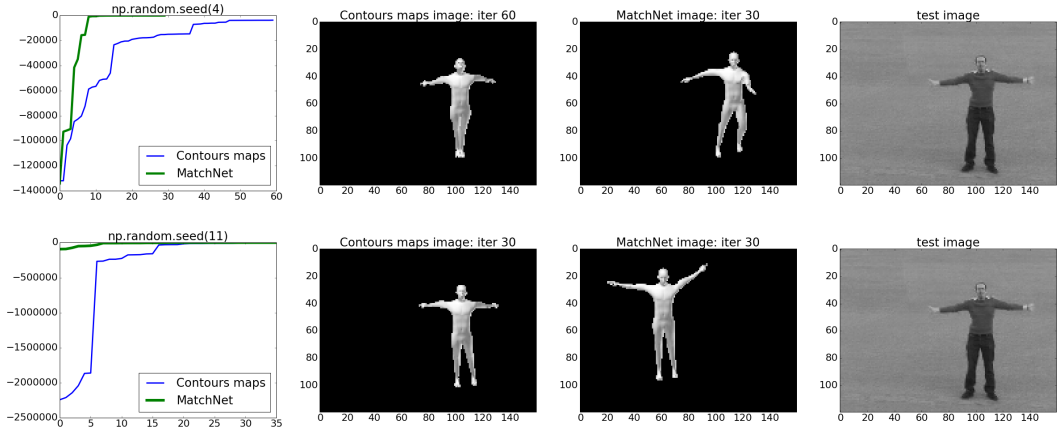


Рис. 8. Карты краев превосходят MatchNet по точности, с MatchNet сходимость быстрее

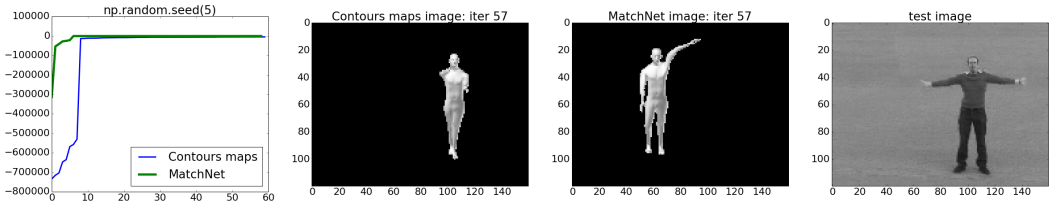


Рис. 9. Модель дает плохой результат в обоих случаях,
с MatchNet сходимость быстрее

6.2 Результаты экспериментов

Всего было выполнено 30 запусков программы с различными k - ядрами генерации случайных чисел, с использованием каждой из функций похожести. Для того, чтобы сравнить графики $Log-likelihood$ для карт краев и MatchNet, был построен усредненный график по 30 запускам программы.

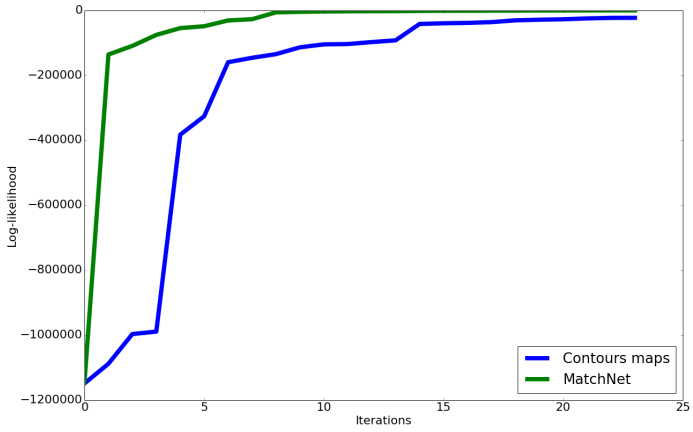


Рис. 9. Графики Log-likelihood, усредненные по 30 запускам программы

По результатам экспериментов можно сделать следующие выводы:

1. Использование l2-метрики в пространстве нейросетевых признаков не увеличивает скорость сходимости модели по сравнению с картой краев

2. **Использование метрики MatchNet увеличивает скорость сходимости модели**

Количество итераций, необходимых для сходимости модели с метрикой MatchNet, меньше по двум основным причинам. Метрика MatchNet увеличивает скорость сходимости порождающей модели по двум основным причинам.

1. Количество итераций, необходимых для сходимости модели с метрикой MatchNet, меньше, так как большее количество гипотез на стадии сэмплирования отклоняется. В отличие от MatchNet, метрика карт краев более чувствительна к незначительным сдвигам позы, и при сэмплировании методом Метрополиса-Гастингса изображения-гипотезы, идущие друг за другом, часто очень незначительно отличаются друг от друга.

2. При этом метрика MatchNet может показывать большую степень близости для изображений, имеющих отличия, чем другие метрики, и поэтому сходимость завершается.

7 Заключение

В ходе курсовой работы были выполнены поставленные задачи:

- Проведен обзор литературы о порождающем подходе и обзор методов задания функции похожести. Выбраны наиболее перспективные методы для использования в порождающих моделях.
- Выбранные методы задания функции похожести были реализованы.
- Было проведено сравнение скорости сходимости модели human pose в зависимости от использования различных функций похожести.

Эксперименты показали, что использование функции похожести, обученной с помощью нейросети MatchNet, увеличивает скорость сходимости порождающей модели по сравнению с ранее использованной метрикой сопоставления карт краев изображений.

Недостатком нейросети MatchNet можно считать медленную скорость извлечения признаков. Несмотря на небольшое число итераций, необходимых для сходимости, время на одну итерацию достаточно велико. Поэтому в дальнейшем имеет смысл провести эксперименты с другими нейросетевыми архитектурами, используемыми для обучения функции похожести. В целом, эксперименты показали, что использование глубинных нейросетей может значительно увеличить скорость сходимости порождающей модели.

Для извлечения нейросетевых признаков имеет смысл, вероятно, использовать нейросети, обученные без учителя, для решения задач сопоставления дескрипторов [7]. И нейросети, специфичные для задачи распознавания позы человека (обученные на соответствующих датасетах), как, например, в [5].

Помимо этого, в дальнейшем необходимо протестировать модель на большем количестве данных.

Список литературы

[1] A. Krizhevsky, I. Sutskever è G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, *Advances in neural information processing systems*, 2012, c. 1097-1105.

[2] Tejas D. Kulkarni (MIT), Pushmeet Kohli (MSR Cambridge, UK), Joshua B. Tenenbaum (MIT),Vikash Mansinghka (MIT), "Picture: A Probabilistic Programming Language for Scene Perception"

[3] John Canny. "A computational approach to edge detection". *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov. 1986.

[4] Lowe D. G. "Object recognition from local scale-invariant features"// *Proc. Intl. Conference on Computer Vision.* – 1999. – P. 1150–1157.

[5] T Pfister, J Charles, A Zisserman. "Flowing convnets for human pose estimation in videos". *Proceedings of the IEEE International Conference on Computer Vision*, 1913-1921

[6] Philipp Fischer, Alexey Dosovitskiy, Thomas Brox. "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT". arXiv pre-print arXiv:1405.5769 (2014)

[7] Dosovitskiy, A., Springenberg, J.T., Brox, T.: "Unsupervised feature learning by augmenting single images pre-print, arXiv:1312.5242v3 (2014)

[8] Sergey Zagoruyko, Nikos Komodakis. "Learning to Compare Image Patches via Convolutional Neural Networks arXiv:1504.03641 [cs.CV] (2015)

[9] Han Xufeng, Leung Thomas, Jia Yangqing, Sukthankar Rahul, Berg Alexander. C., "MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching *CVPR*, 2015

[10] Karen Simonyan, Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition arXiv:1409.1556 [cs.CV], 2014

[11] Y. Jia. Caffe: An open source convolutional archi- tecture for fast feature embedding. http://caffe. berkeleyvision.org/, 2013.

[12] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, Hod Lipson. "Understanding Neural Networks Through Deep Visualization *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015