Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем
Кафедра системного программирования

Лучинский Владимир Дмитриевич

Распознавание городских объектов при различных погодных условиях

Курсовая работа

Научный руководитель:
ст. преп. Смирнов М. Н.

Консультант:
Пенкрат Н. А.

Санкт-Петербург
2019

Оглавление

Введение

Что если, захотев перекусить в каком-то кафе, которое видим в первый раз, мы могли бы по фотографии заведения понять, что это за кафе, узнать отзывы посетителей или фирменные блюда, которые там подают? Это возможно реализовать, решив задачу распознавания, то есть, получив на вход фотографию объекта, понять, что за здание на ней изображено, и выдать интересующую нас информацию о нём. В данной работе предлагается использовать для этого глубокую нейронную сеть (deep neural network).

Чтобы натренировать глубокую сеть, нам в первую очередь необходимо достаточное количество размеченных тренировочных данных (датасет). Недостаток тренировочных данных приведёт к переобучению (overfitting), и наша модель будет показывать прекрасные результаты на тех данных, которые были ей представлены в процессе обучения, но на примерах, не участвовавших в обучении (на примерах из тестовой выборки), будет работать относительно плохо.

Одним из подходов к решению данной проблемы является использование различных регуляризационных техник. В последнее время были предложены подходы, которые успешно применяются в обучении глубоких нейронных сетей. В частности, техники исключения (dropout)[4] и нормализации батчей (batch normalization)[12], хорошо зарекомендовавшие себя на практике.

Кроме того, мы можем бороться с переобучением с помощью аугментации. Аугментация данных — это процесс дополнения датасета сходными данными, созданными из уже имеющихся. Аугментация повсеместно применяется при работе с изображениями и часто включает в себя поворот, смещение, замутнение и другие модификации существующих изображений, что позволяет нейросети выделять их наиболее важные признаки[19].

Помимо прочего, если мы хотим использовать нейронную сеть в изменяющихся погодных условиях, то нам необходимо иметь достаточное количество примеров для каждого погодного явления, в котором

мы будем применять нашу сеть. К примеру, мы не сможем показать вразумительный результат на зимних данных, обучившись только на летних. Это серьезно усложняет сбор датасета и повышает временные затраты на его сбор до почти невозможных значений.

Для снижения стоимости сбора необходимых нам данных, нужно уметь генерировать их, имея только образцы, полученные, например, в летних и ясных условиях.

Это задачу можно сформулировать, как перевод одного изображения в другое (image-to-image translation)[9], то есть превращение одного представления какого-то объекта, x, в другое, y. Например, преобразовать дневное фото в ночное, зимнее — в летнее, черно-белое изображение — в цветное. В последние годы, для решения этой задачи успешно применяются генеративные соревновательные сети (generative adversarial networks, GAN)[6], генерирующие изображения, практически неотличимые от реальных[9, 21, 17].

1. Постановка задачи

Цель работы — разработать методику, позволяющую распознавать городские объекты как летом, так и зимой, и проверить пользу применения генеративных сетей в решении данной задачи.

Итак, для того чтобы подтвердить работоспособность предложенного подхода, необходимо:

1. Определиться с архитектурой генеративной нейронной сети(GAN), которая будет генерировать зимние изображения из летних, и летние из зимних.

2. Определиться с архитектурой нейронной сети, распознающей городские объекты (обозначим её T).

3. Выбрать стратегию аугментации (обозначим её A) для обучения T.

4. Выбрать или собрать датасет (обозначим его $d1$) для обучения GAN.

5. Реализовать GAN или найти готовую реализацию.

6. Обучить GAN.

7. Выбрать или собрать датасет (обозначим его $d2$) для обучения T.

8. Расширить $d2$, используя обученную сеть GAN.

9. Реализовать A или найти готовую реализацию.

10. Использовать стратегию A и датасет $d2$ для обучения T.

Вообще говоря, можно выбрать любое преобразование погодных условий, но в рамках данной работы принято решение работать с вышеназванными (зима $\rightarrow$ лето, лето $\rightarrow$ зима), так как подобные преобразования уже успешно проделывались в ряде работ [21, 17].

Кроме того, стоит уточнить, что в тестовой выборке $d2$ должны содержаться изображения одних и тех же объектов в различные времена года, чтобы продемонстрировать повышение точности локализации.

Также стоит заметить, что композицию из GAN и аугментационного слоя можно применять при решении других задач распознавания в естественных условиях в качестве инструмента расширения датасета.

2. Обзор

2.1. Генеративные соревновательные сети (Generative adversarial networks)

2.1.1. Принцип работы

В 2014 году, Гудфеллоу (Goodfellow) и др. опубликовали статью [6], в которой описали метод обучения генеративных моделей, названных генеративными соревновательными нейросетями (Generative Adversarial Networks, далее GAN). По сути GAN — это две различные сети.

Первая — традиционная свёрточная нейронная сеть, которая называется дискриминатором. Мы учим дискриминатор классифицировать входные изображения на реальные (принадлежащие обучающей выборке) и ложные (не представленные в обучающей выборке).

Другая сеть, называемая генератором, принимает на вход случайный шум и преобразует его в изображение, используя слои деконволюции (deconvolution, их также называют upconvolution, transpose convolution)[5]. Цель генератора — обмануть дискриминатор так, чтобы он принимал сгенерированные изображения за настоящие.

Мы можем выразить попытки генератора (G) обмануть дискриминатор (D) и стремление D правильно классифицировать настоящие и искусственные изображения в виде минимакса (теория игр) :

$$\min_G \max_D \; E_{x\sim p_{data}} \left[\log D(x)\right] + E_{z\sim p(z)} \left[\log\left(1 - D(G(z))\right)\right], \qquad (1)$$

где $x \sim p_{data}$ — это образцы входных данных, $z \sim p(z)$ — случайный шум, $G(z)$ — сгенерированные сетью G изображения, а $D(x)$ — результат работы дискриминатора, представляющий собой вероятность того, что данные, полученные на вход — настоящие.

Минимизация выражения (1) заключается в том, чтобы поочерёдно делать шаги градиентного спуска для сетей G и D:

1. Обновить параметры генератора G, чтобы минимизировать вероятность того, что дискриминатор D правильно классифицирует

изображение.

2. Обновить параметры дискриминатора D, чтобы максимизировать вероятность того, что дискриминатор не ошибётся.

Стоит заметить, что на практике соотношение (1), как правило, не используется. В той же оригинальной статье 2014 года, было предложено обучать генератор, максимизируя вероятность ошибки дискриминатора. Это небольшое изменение помогает решить проблему затухающих градиентов в тех случаях, когда вероятность выдаваемая дискриминатором близка к единице.

2.1.2. CycleGAN

CycleGAN — представляет собой GAN, решающий задачу отображения одного пространства изображений в другое (image-to-image translation). Как правило, для решения подобной задачи нам необходимы пары образцов тренировочных данных $\{x_i, y_i\}_{i=1}^{N}$, $x_i \in X$, $y_i \in Y$, где x_i соответствует y_i, X и Y пространства изображений (см. рис 1). Это накладывает дополнительные требования на датасет.

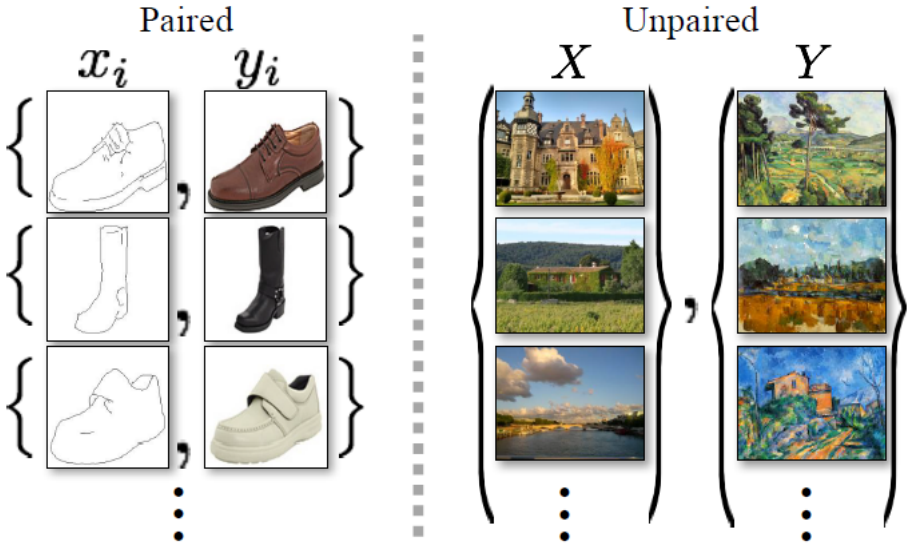

Рис. 1: Отличие парных данных от непарных.
Изображение взято из [21].

Отличительная черта CycleGAN заключается в том, что ему требуются лишь множества $\{x_i\}_{i=1}^{N}$ и $\{y_i\}_{i=1}^{M}$, причем конкретные соотношения между x_i y_i могут быть и неизвестны. Пусть у нас есть два генератора, которые задают отображения G и F на пространствах изображений X, Y, и $G : X \to Y$, $F : Y \to X$. Тогда авторы статьи предполагают, что G должно быть обратным отображением для F, и наоборот, причём функции — биективны.

Выполнение этого предположения достигается за счёт обучения двух генераторов одновременно (каждому из них соответствует свой дискриминатор) и использованием особой функции потерь, которая стимулирует генераторы к тому, чтобы $F(G(x)) \approx x$ и $G(F(y)) \approx y$. Авторы статьи называют её cycle-consistent loss, откуда и происходит название всей сети.

2.1.3. Unsupervised Image-to-Image Translation Networks (UNIT)

UNIT[17] — решает ту же задачу, что и CycleGAN, и также позволяет обучаться на данных, которые не разбиты на пары (см. рис 1). Отличие от CycleGAN заключается в использовании не только GAN, но и генеративных сетей архитектуры VAE[13, 15]. Это позволяет авторам превзойти показатели CycleGAN, однако делает процесс обучения сети более долгим и повышает требования к оборудованию, на котором мы хотим обучать UNIT.

2.2. Аугментация данных

2.2.1. Виды аугментации

Как правило, различают два вида аугментации. Первый используется независимо от того, как размечены данные (то есть без учёта лейбла). Например, добавление к изображению случайного шума, поворот или зеркальное отражение. Такая функциональность встроена во

многие фреймворки для глубокого обучения, например в Keras[1]. Однако, подобные преобразования не должны использоваться вслепую. Если взять, например, MNIST[2] (известный датасет рукописных чисел) и применить к данным из него поворот на 180°, то обучаемая нами модель будет не в состоянии отличить «6» от «9».

Второй тип аугментации данных предполагает трудоемкий процесс увеличения количества данных за счёт смешения различных образцов данных с одинаковым лейблом, чтобы сгенерировать новый образец с таким же лейблом. В данном случае успех, как правило, полностью зависит от интуиции и опыта исследователя, который определяет, какие образцы смешивать и сколько их брать.

Однако, вместо того, чтобы пытаться самим придумать стратегию аугментации, которая приведёт к улучшению текущих результатов обучаемой нейросети (назовём её исходной), мы можем использовать для этого ещё одну модель нейронной сети. Эта модель будет генерировать смешанные образцы таким способом, чтобы повысить показатели конкретной сети (в нашем случае, исходной), используя алгоритм «умной» аугментации, описанный ниже.

2.2.2. Smart Augmentation

«Умная» аугментация (smart augmentation)[16] - это процесс изучения нейронной сетью (сеть A) лучшей аугментационной стратегии для обучения другой нейронной сети (сеть B) для решения конкретной задачи. Результат работы сети A используется в качестве входных данных для сети B.

Сеть A принимает на вход несколько тренировочных образцов одного из классов датасета и генерирует новый образец того же самого класса, который позволяет снизить значение функции потери сети B. В качестве примера рассмотрим рис. 2. Изображение слева (обозначим его за I) получено смешением изображений справа обученной сетью A.

[1]F. Chollet, “Keras,” https://github.com/fchollet/keras, 2015

[2]LeCun Yann, Cortes Corinna, MNIST handwritten digit database, http://yann.lecun.com/exdb/mnist, 2010

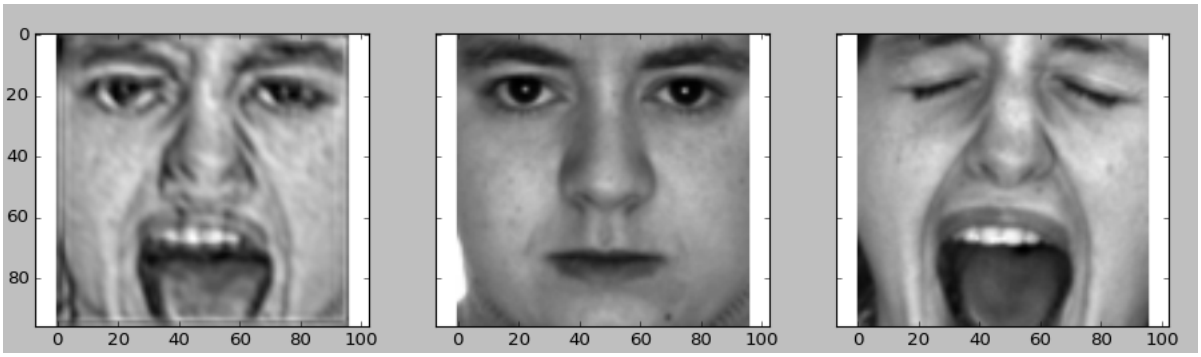

Рис. 2: Смешение изображений. Изображение взято из [16].

Изображение I не является фотореалистичным, однако оно помогает нейросети B, которая решает задачу классификации мужчин и женщин по фотографии, лучше выделять свойства, характерные тому или иному классу.

По своей архитектуре нейронная сеть A представляет генеративную сеть с теми отличиями, что на неё влияет сеть B во время обратного прохода (back propagation step), и нейросеть A принимает на вход одновременно несколько изображений одного класса, а не одно. Это заставляет данные, порождённые нейронной сетью A, сходиться к тем, которые больше всего снизят ошибку сети B на конкретной задаче, которую та решает. Кроме того, генеративная сеть A также контролируется собственной функцией потерь, чтобы результат её работы был достаточно близок к другим представителям класса, образцы которого подаются ей на вход.

Стоит заметить, что конкретную задачу здесь решает сеть B, однако архитектура сети A от этой задачи не зависит.

2.3. Свёрточные нейронные сети

2.3.1. Остаточная нейронная сеть (Residual neural network, ResNet)

Остаточная нейронная сеть (Residual neural network)[2] представляет из себя свёрточную нейронная сеть, использующую непоследовательные соединения (skip-connections), чтобы пропускать некоторые слои,

причем такое соединение не пропускает больше одного слоя. Это позволяет сетям архитектуры ResNet улучшить поток градиентов через слои, и тем самым увеличить их количество, что повышает результаты (см. рис 3), показываемые моделью при решении различных задач компьютерного зрения.

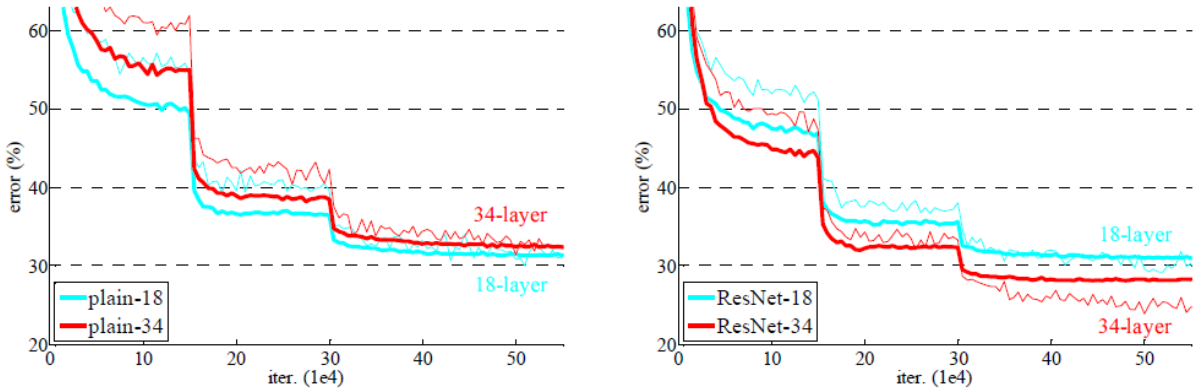

Рис. 3: Сравнение функции ошибок сетей архитектуры ResNet с сетями не использующими непоследовательные соединения (skip-connections) при обучении на ImageNet[10]. Тонкие кривые обозначают ошибку при обучении, толстые — при валидации.
Изображение взято из [2].

2.3.2. Нейронная сеть Inception

Свёрточная нейронная сеть Inception[7] — это сеть, состоящая из так называемых Inception модулей (см рис. 4). Их основной особенностью является применение свёрток размера 1x1 перед «дорогими» свёртками больших размеров. Это позволяет снизить количество проводимых операций и ускорить процесс обучения сети, что делает сеть довольно эффективной. На данный момент последней её версией является Inception-v4[11].

2.4. Распознавание объектов

Одним из самых эффективных подходов, применяющихся для распознавания различных объектов, например лиц, является изучение свойств (feature learning) этих объектов[22]. Данные свойства представляют из

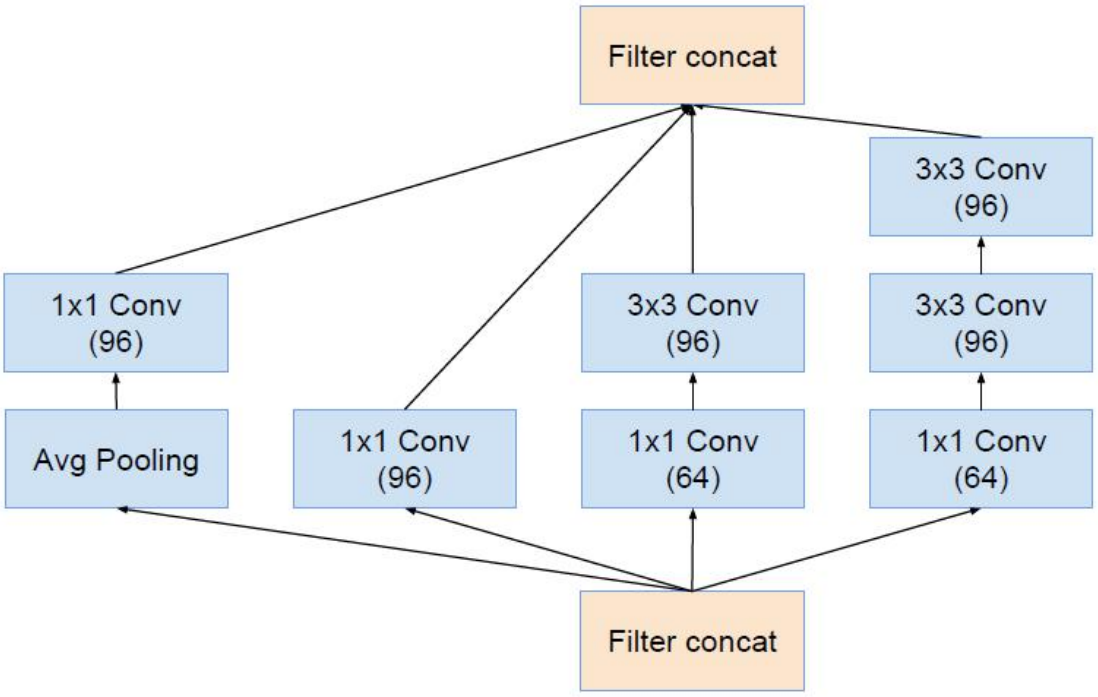

Рис. 4: Строение модуля в Inception-v4. Изображение взято из [11].

себя вектора в n-мерном евклидовом пространстве (поэтому далее они также будут называться характеристическими векторами, а само пространство - пространством свойств). Свёрточные нейросети, применяющиеся для этого, выполняют выделение свойств и предсказание лейбла, сопоставляя входные данные сначала «глубоким» свойствам (выходные данные последнего свёрточного слоя), а затем предсказанным лейблам (классам входных данных).

В распознавании объектов, например лиц или зданий, классы возможных тестовых данных также принадлежат тренировочному датасету. Таким образом, мы можем подходить к распознаванию, как к задаче классификации, и использовать для обучения категориальную кросс-энтропию(categorical cross-entropy loss, softmax loss) в качестве функции потерь.

После обучения, получая на вход изображение объекта, с помощью нейронной сети ему сопосталяется характеристический вектор, затем сравнивается по норме с соответствующими векторами изображений в датасете и самый близкий считается самым похожим.

Для того чтобы расстояния между такими векторами были малы для представителей одного класса и велики для остальных классов, используется центральная потеря (center loss)[3]. Её идея заключается в изучении центра каждого класса в пространстве свойств и минимизации расстояния изображения из этого класса до его центра.

3. Описание решения

3.1. Архитектура

Архитектура решения состоит из трёх главных компонент:

1. Генеративная соревновательная нейронная сеть CycleGAN.
2. Генеративная сеть, реализующая алгоритм Smart Augmentation.
3. Распознающая свёрточная нейросеть.

Принцип работы заключается в том, что сначала мы обучаем CycleGAN переводить летние изображения в зимние и используем его для пополнения нашего датасета. Далее проводятся эксперименты: обучаем распознающую сеть на исходном датасете и на расширенном. Кроме того, во время тренировки ей на вход также подаются смешанные с помощью «умной» аугментации летние и зимние изображения. Такая аугментация данных призвана в купе с расширением датасета призвана побудить распознающую сеть выделять независящие от сезона свойства городских объектов.

Распознающая сеть представляет из себя свёрточную нейросеть. На этапе обучения на последний её слой дополнительно напущен слой регрессии, чтобы результат работы был скаляром, выражающим уверенность сети в том, что образец данных, поданный на вход, принадлежит тому или иному классу, то есть решается задача классификации.

После тренировки она принимает на вход изображение и вычисляет для него характеристический вектор (выход последнего слоя) и находит ближайший к нему по норме вектор из тех, что были получены из уже имеющихся в базе изображений. Таким образом, мы получаем городской объект «самый похожий» на входной.

3.2. Обоснование принятых технических решений

Важно заметить, что реализация первых двух компонентов архитектуры не зависит от третьего, их можно использовать со свёрточ-

ной нейросетью любой архитектуры. Поэтому при выборе распознающей сети, которая будет осуществлять изучение особых свойств (feature learning), было решено сделать выбор в пользу свёрточной сети архитектуры Inception-Resnet-v1, которая является сочетанием сетей Inception и Resnet и способна показывать хорошие результаты на широком круге задач[11]. В качестве функции ошибки для неё выбрана категориальная кросс-энтропия (softmax), которая способна сходиться на небольших датасетах, а также центральная функция потери (center loss), побуждающая изучаемые свойства быть хорошо разделимыми и входящая в число самых эффективных фукций ошибки для распознавания лиц[22].

Стоит отметить, что в работе используется готовая реализация сети Inception-Resnet-v1, найденная в репозитории Facenet[3], поэтому далее она также будет именоваться Facenet.

Для преобразования погодных условий был выбран CycleGAN, так как датасет для обучения довольно легко собрать (можно использовать изображения, не разбитые на пары), и в отличие от альтернативных решений, которые способны добиваться фотореалистичного изменения[17], его проще обучать на имеющемся оборудовании (видеокарты NVIDIA: GeForce GTX 1080 Ti, GeForce GTX 1060).

Альтернативой применению «умной» аугментации является простое добавление данных в датасет для обучения локализационной сети. Планируется протестировать как один, так и другой подход, однако предполагается, что использование аугментации позволит нам показать лучшие результаты относительно наивного подхода.

Для реализации алгоритма смарт аугментации (Smart Augmentation) был выбран фреймворк глубокого обучения TensorFlow[20]. Выбор проводился между ним и фреймворком Pytorch[1], однако предпочтение было отдано TensorFlow, который обладает широким сообщество программистов, является более зрелым, чем Pytorch, а также имеет средства для развертывания на мобильных устройствах, на которых в перспективе будет применяться полученная модель нейронной сети.

[3]https://github.com/davidsandberg/facenet

3.3. Обучение GAN

Генеративные соревновательные нейронные сети, к которым относится CycleGAN, на данный момент показывают лучшие результаты среди других типов генеративных сетей, однако они же самые трудные в обучении[8]. Так как в GAN дискриминатор и генератор соревнуются друг с другом, то их функции ошибки постоянно колеблются в некоторых пределах, и результаты обучения часто приходиться оценивать просто анализируя сгенерированные изображения.

Для обучения данной сети было решено использовать данные, собранные вручную (1000 летних и 1000 зимних изображений). Так как их одних довольно мало для того, чтобы натренировать GAN, то дополнительно используется датасет SYNTHIA[18], который представляет из себя набор синтетических изображений городских сцен, сделанных в разное время года (примерно 7000 изображений для каждого сезона). Во время тренировки синтетические данные использовались с настоящими в соотношении 2:1 (т.е. из 7000 изображений случайным образом выбирались 2000). Гиперпараметры применялись те же, что использовали авторы данной нейросети, за исключением тренировочного батча, который был увеличен до пяти изображений, чтобы ускорить обучение, которое всего заняло 40 тысяч итераций.

Следует заметить, что полученная модель работает не идеально, однако целью данной работы является не фотореалистичное преобразование изображений, а улучшение распознавания в зимних условиях. Несмотря на это, обученная модель лучше подходит для решения данной задачи чем готовая модель авторов CycleGAN (см рис. 5 и 6), так как та просто напросто была натренирована работать с фотографиями дикой природы и при генерации изображений не сохраняет исходный цвет здания, что может быть критично для проекта. Стоит отметить, что создатели нейронной сети UNIT не выложили свою обученную модель в открытый доступ, поэтому провести сравнение с ней оказалось невозможным.

Рис. 5: Полученная модель

Рис. 6: Готовая модель авторов CycleGAN

3.4. Реализация Smart Augmentation

Так как алгоритм смарт аугментации заключается в применении небольшое свёрточной сети для смешивания изображений (см рис. 7), далее это сеть также будет называться смарт аугментацией. Одной из составляющих функции потерь данной нейросети является функция потерь распознающей сети(L_B), поэтому они обучаются одновременно.

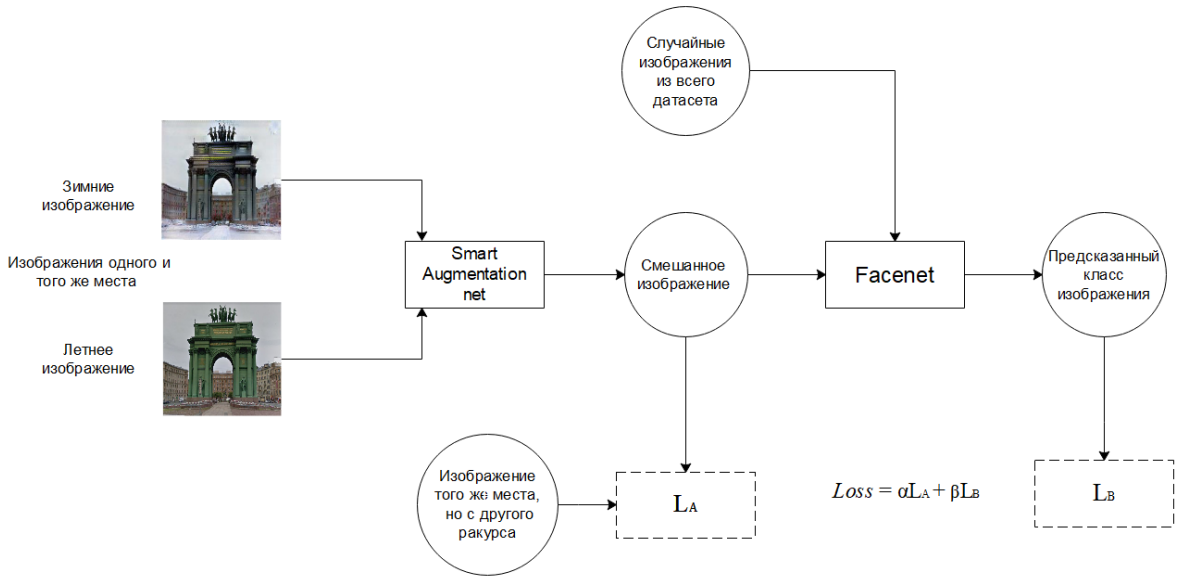

Рис. 7: Диаграмма потока данных сети Smart Augmentation

Основным принципом работы фреймворка TensorFlow является построение статического вычислительного графа(computational graph) и его последующее применение для обучения сети, поэтому для реализации смарт аугментации необходимо было сначала изучить граф исходной сети Facenet, показатели которой мы хотим улучшить.

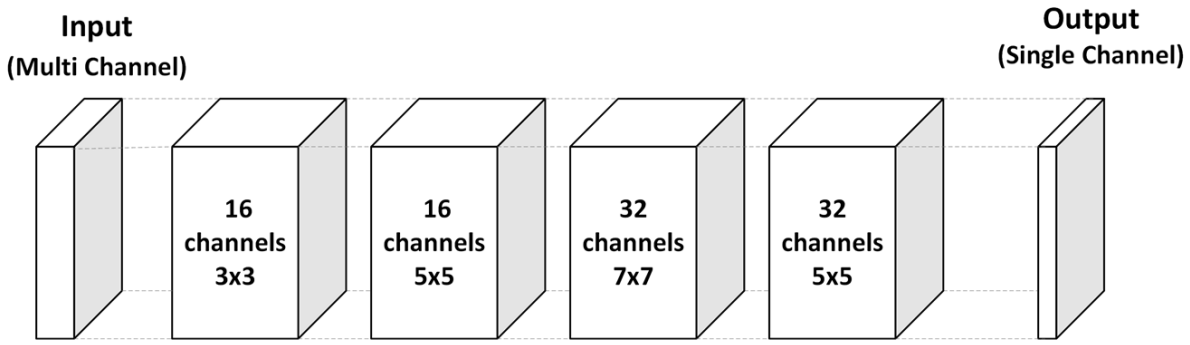

Рис. 8: Строение сети Smart Augmentation. Изображение взято из [16].

Затем была написана сама сеть Smart Augmentation и встроена в Facenet[4]. Причем старые узлы исходного графа не подверглись изменениям, поэтому компоненты, измеряющие показатели, совместимы с моделями, которые получаются при обучении с использованием смарт аугментации.

Основной задачей при написании сети Smart Augmentation была реализация потока данных(data flow), необходимых для её работы (см рис. 7). Нейросеть работает с тройками изображений. Два из них(I_1, I_2) - это изображения одного и того же здания, одно из которых является фотографией в летних условиях, а другое получено с помощью обученной модели CycleGAN. Данные два трёхканальных изображения соединяется с друг другом в измерении, представляющем количество каналов, и передаются на вход смарт аугментации. Сеть Smart Augmentation пропускает полученный тензор через свёрточные слои(см рис 8), сохраняя ширину и высоту данных, но уменьшая количество снова до трёх, и выдаёт смешанное таким образом изображение(O). Третье изображение(I_3) представляет из себя случайное летнее изобра-

[4]Код представлен в репозитории https://github.com/ucLh/facenet/tree/smart_aug (папка smaug и файл train_softmax_w_smaug.py в папке src).

жение из того же класса, что и первые два, однако сделанное с другого ракурса.

Функция ошибки смарт аугментации состоит из двух слагаемых: L_B, которая упомяналась выше, и L_A, которая представляет из себя среднеквадратичную функцию потерь, вычисляющуюся от разности I_3 и O, предназначенную для обеспечения близости полученного с помощью сети изображения и другими представителями класса. L_A и L_B смешиваются с коэффициентами 0.7 и 0.3 соответственно. Выбор коэффициентов обусловлен результатами, полученными авторами Smart Augmentation.

3.5. Эксперименты

Для проверки концепции (обозначим её за H) применения сгенерированных с помощью GAN изображений для повышения точности распознавания городских объектов в различных погодных условиях было проведено несколько экспериментов.

Во всех экспериментах применялся оптимизатор Adam[14], с довольно высоким стартовым коэффициентом скорости обучения (learning rate) равным 0.005 для более быстрого схождения, который затем уменьшался в 2 раза всякий раз, когда изменения функции ошибки становились незначительными. Также стоит отметить, что для обучения распознающей нейронной сети использовалась центральная функция ошибки (center loss)[3] с коэффициентом 0.5, который выбран на основании результатов авторов подхода.

Для обучения применялся датасет, собранный вручную с помощью поисковой системы Google, содержащий 20 146 изображений и 630 классов в тренировочной выборке. Тестовый датасет D состоит из 2255 изображений и включает в себя только летние фотографии объектов, его подмножество, S, содержит 316 изображений для 30 зданий. К сожалению, на момент написания собрать реальные зимние изображения (всего 568 фотографий) удалось только для этих 30 зданий.

Распознающая была обучена как с применением смарт аугментации

и зимних изображений, полученных с помощью CycleGAN, так и без. Из экспериментов в таблицах 1 и 3 видно, что на летних данных точность (accuracy) распознающей сеть возросла на 5 %, а на зимних - на 11%, что подтверждает концепцию H, ибо улучшения заметны как на зимних, так и на летних данных. Кроме того, стоит отметить, применение Smart Augmentation также дало некоторый прирост точности, относительно экспериментов, представленных в таблице 2.

№	Датасет	Потеря	Кросс-энтропия	Точность
1	D	5.218	1.285	0.743
2	S	4.436	0.510	0.861
3	W	6.751	2.850	0.423

Таблица 1: Результаты сети, обученной с использованием GAN и Smart Augmentation

№	Датасет	Потеря	Кросс-энтропия	Точность
4	D	6.251	1.337	0.722
5	S	5.469	0.558	0.854
6	W	7.905	3.024	0.403

Таблица 2: Результаты сети, обученной с использованием GAN, но без Smart Augmentation

№	Датасет	Потеря	Кросс-энтропия	Точность
7	D	6.954	1.456	0.692
8	S	6.193	0.696	0.813
9	W	9.024	3.553	0.315

Таблица 3: Результаты сети, обученной без использования GAN и Smart Augmentation

3.6. Возможные пути улучшения и развития

Хотя распознавание зимних объектов было улучшено, оно всё ещё довольно сильно отстает от результатов на летних изображениях. Од-

ним из возможных путей улучшения может послужить использование для обучения GAN тренировочных данных распознающей сети, а также экспериментирование с различными архитектурами свёрточных сетей и функциями потерь для неё.

Также стоит заметить, что результаты на выборке S значительно выше, чем на D. Вероятно, это связано с тем, что в на первые 600 классов в среднем приходится 28,5 изображений, а на последние 30 - примерно 100 изображений, и при расширении первых классов датасета можно рассчитывать на прирост точности.

Кроме того, одним из очевидных путей развития является применение генеративных сетей для преобразования летних изображений в изображения других, отличных от зимнего, сезонов.

Заключение

В соответствии с постановкой задачи проделано следующее:

1. Для генерации зимних изображений из летних решено использовать GAN архитектуры CycleGAN[21].
2. Для распознающей сети выбрана нейронная сеть Facenet[5].
3. Выбрана стратегия аугментации Smart Augmentation[16].
4. Решено, какие данные использовать для обучения GAN.
5. Найдена реализация CycleGAN[6], которую планируется использовать в данной работе.
6. Реализован алгоритм Smart Augmentation.
7. Модель CycleGAN обучена преобразовывать летние изображения в зимние.
8. Произведено обучение распознающей сети.
9. Выполнен анализ полученных результатов.
10. Подтверждена польза применения GAN для улучшения распознавания объектов в зимних условиях.

[5]https://github.com/davidsandberg/facenet

[6]https://github.com/LynnHo/CycleGAN-Tensorflow-PyTorch

Список литературы

[1] Automatic differentiation in PyTorch / Adam Paszke, Sam Gross, Soumith Chintala et al. // NIPS-W. — 2017.

[2] Deep Residual Learning for Image Recognition / Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun // CoRR. — 2015. — Vol. abs/1512.03385. — 1512.03385.

[3] A Discriminative Feature Learning Approach for Deep Face Recognition / Yandong Wen, Kaipeng Zhang, Zhifeng Li, Yu Qiao. — Vol. 9911. — 2016. — 10. — P. 499–515.

[4] Dropout: A Simple Way to Prevent Neural Networks from Overfitting / Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky et al. // Journal of Machine Learning Research. — 2014. — 06. — Vol. 15. — P. 1929–1958.

[5] Dumoulin Vincent, Visin Francesco. A guide to convolution arithmetic for deep learning // arXiv e-prints. — 2016. — . — P. arXiv:1603.07285. — 1603.07285.

[6] Generative Adversarial Nets / Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza et al. // Advances in Neural Information Processing Systems 27 / Ed. by Z. Ghahramani, M. Welling, C. Cortes et al. — Curran Associates, Inc., 2014. — P. 2672–2680. — URL: `http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf`.

[7] Going Deeper with Convolutions / Christian Szegedy, Wei Liu, Yangqing Jia et al. // CoRR. — 2014. — Vol. abs/1409.4842. — 1409.4842.

[8] Goodfellow Ian J. NIPS 2016 Tutorial: Generative Adversarial Networks // CoRR. — 2017. — Vol. abs/1701.00160. — 1701.00160.

[9] Image-to-Image Translation with Conditional Adversarial Networks / Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros // CoRR. — 2016. — Vol. abs/1611.07004. — 1611.07004.

[10] ImageNet Large Scale Visual Recognition Challenge / Olga Russakovsky, Jia Deng, Hao Su et al. // International Journal of Computer Vision (IJCV). — 2015. — Vol. 115, no. 3. — P. 211–252.

[11] Szegedy Christian, Ioffe Sergey, Vanhoucke Vincent, Alemi Alex. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. — 2016. — 1602.07261.

[12] Ioffe Sergey, Szegedy Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift // CoRR. — 2015. — Vol. abs/1502.03167. — 1502.03167.

[13] Jimenez Rezende Danilo, Mohamed Shakir, Wierstra Daan. Stochastic Backpropagation and Approximate Inference in Deep Generative Models // arXiv e-prints. — 2014. — . — P. arXiv:1401.4082. — 1401.4082.

[14] Kingma Diederik, Ba Jimmy. Adam: A Method for Stochastic Optimization // International Conference on Learning Representations. — 2014. — 12.

[15] Kingma Diederik P, Welling Max. Auto-Encoding Variational Bayes // arXiv e-prints. — 2013. — . — P. arXiv:1312.6114. — 1312.6114.

[16] Lemley Joseph, Bazrafkan Shabab, Corcoran Peter. Smart Augmentation - Learning an Optimal Data Augmentation Strategy // CoRR. — 2017. — Vol. abs/1703.08383. — 1703.08383.

[17] Liu Ming-Yu, Breuel Thomas, Kautz Jan. Unsupervised Image-to-Image Translation Networks // CoRR. — 2017. — Vol. abs/1703.00848. — 1703.00848.

[18] The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes / German Ros, Laura Sellart, Joanna Materzynska et al. // The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — 2016. — June.

[19] Simard P. Y., Steinkraus D., Platt J. C. Best practices for convolutional neural networks applied to visual document analysis // Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. — 2003. — Aug. — P. 958–963.

[20] Abadi Martín, Agarwal Ashish, Barham Paul et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. — 2015. — Software available from tensorflow.org. URL: `http://tensorflow.org/`.

[21] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks / Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A Efros // Computer Vision (ICCV), 2017 IEEE International Conference on. — 2017.

[22] Wang Mei, Deng Weihong. Deep Face Recognition: A Survey // CoRR. — 2018. — Vol. abs/1804.06655. — 1804.06655.