МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики

Кафедра дискретной математики и алгоритмики

Регрессия 2D ключевых точек лица на основе нейронных сетей

Курсовая работа

Тылецкого Арсения Витальевича
студента 3 курса
специальность "информатика"

Научный руководитель:
старший преподаватель Д. И. Пирштук

Минск, 2023

РЕФЕРАТ

Курсовая работа, 33 стр., 14 иллюстр., 7 табл., 11 источников.

Ключевые слова: РЕГРЕССИЯ ТОЧЕК ЛИЦА, НЕЙРОННЫЕ СЕТИ, RESNET, MOBILENETV2, EFFICIENTNET, MOBILENETV3, MOBILEVIT, ФУНКЦИЯ ПОТЕРЬ WING, АУГМЕНТАЦИЯ, LAPA.

Объекты исследования – задача регрессии ключевых точек лица, архитектуры нейронных сетей.

Цель исследования – сравнение архитектур нейронных сетей и изучение влияния различных аспектов в общем алгоритме обучения нейронных сетей на задаче регрессии 2D ключевых точек лица.

Методы исследования – системный подход, изучение соответствующей литературы и электронных источников, постановка задачи и её решение.

В результате исследования рассмотрены различные архитектуры нейронных сетей, как классические, такие как ResNet и MobileNetV2, так и более современные – EfficientNet, MobileNetV3, MobileViT, произведено их сравнение на задаче регрессии ключевых точек лица, выявлено влияние различных аспектов в общем алгоритме обучения нейронных сетей.

Области применения – редактирование лиц, виртуальная реконструкция лица, распознавание эмоций, отслеживание взгляда водителя для мониторинга внимания, бьютификация лиц, применение масок лица.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Распознавание ключевых точек лица является одним из основополагающих элементов общего процесса обработки лица. Основными областями применения являются редактирование лиц и дополненная реальность. Например, распознавание ключевых точек лица используется для виртуальной реконструкции лица, распознавания эмоций, отслеживание взгляда водителя для мониторинга внимания, бьютификации лиц и т.д.. Ранние подходы были пригодны только для обнаружения точек в контролируемых условиях, чего явно недостаточно. Ансамбли решающих деревьев, а позже и нейронные сети, продемонстрировали поразительное улучшение качества в решении задачи распознавания ключевых точек лица "в дикой природе" (in-the-wild), и в настоящее время они изучаются многими специалистами в этой области.

Все более актуальными становятся быстрые алгоритмы регрессии ключевых точек лица, позволяющие определять точки в реальном времени. Наиболее популярным примером из данной области являются маски. Маски используют многие приложения, особенно мобильные. Например, такие гиганты как Instagram, VK, Snapchat, MSQRD. Чтобы применить маску достаточно применить к ней простейшие геометрические преобразование, исходя из полученных ключевых точек лица. Существуют и более сложные варианты 3D масок.

Идею регрессии ключевых точек лица можно перенести и на регрессию ключевых точек тела человека. Отслеживания ключевых точек человеческого тела нашло свое применение в компьютерных играх. Этому также поспособствовало развитие камер глубины, которые снимают видео, в каждом пикселе которого хранится не цвет, а расстояние до объекта в этой точке. Революцию области отслеживания ключевых точек тела в реальном времени произвел запуск устройства Kinect, разработанным компанией Microsoft для своей игровой консоли Xbox.

В данной работе мы сначала строго поставим задачу нахождения ключевых точек лица, рассмотрим метрики оценивания точности решения задачи, рассмотрим наиболее популярные датасеты для данной задачи. Затем мы произведем краткий обзор методов решения поставленной задачи. После чего будет рассмотрен процесс подготовки данных используемого нами датасета, описан общий алгоритм обучения нейронных сетей для задачи регресии ключевых точек лица. Мы рассмотрим различные архитектуры нейронных сетей, как классические, такие как ResNet и MobileNetV2, так и более современные – EfficientNet, MobileNetV3, MobileViT. Будет произведено сравнение данных архитектур на задаче регрессии ключевых точек лица, а также выявлено влияние различных аспектов в общем алгоритме обучения нейронных сетей.

ГЛАВА 1
ПОСТАНОВКА ЗАДАЧИ НАХОЖДЕНИЯ КЛЮЧЕВЫХ ТОЧЕК ЛИЦА И ЕЕ ПРИМЕНЕНИЕ

1.1 Постановка задачи нахождения ключевых точек лица

Обозначим через I входное изображение, которое представлено в виде 3-мерного тензора размера $W \times H \times C$, где W, H, C - ширина, высота и количество цветовых каналов изображения соответственно. Обычно цветные изображения используют 3 канала: красного, зеленого и синего цветов. Пусть также $x_i \in \mathbb{R}^2$ есть x, y координаты i-ой ключевой точки изображения I. Тогда через вектор $S = (x_1^T, x_2^T, \ldots, x_p^T)^T \in \mathbb{R}^{2p}$ можно обозначить все p ключевых точек лица на изображении I. Вектор S в дальнейшем будем называть формой лица (shape). Задача обнаружения ключевых точек лица состоит в том, чтобы найти такую функцию $f : I \to S$, которая по входному изображению I предсказывает вектор ключевых точек лица $\hat{S}$. Количество точек лица p, а также точное отображение i-ой точки лица в ее координаты на картинке заданы в датасете. Пример размеченных ключевых точек лица приведен на рисунке 1.1.

Рисунок 1.1 — Пример размеченных ключевых точек лица

1.2 Метрики точности регрессии

Естественно, для оценки качества алгоритма необходимо ввести метрики оценки точности. Вообще говоря, метрики также могут определяться на уровне датасета, чтобы можно было сравнивать алгоритмы в рамках одного датасета по общему критерию. Однако наиболее популярной метрикой в решении задачи регрессии ключевых точек лица является NME (Normalized Mean Error), которая определяется следующим образом:

$$NME = \frac{1}{K}\sum_{k=1}^{K} NME_k = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{p}\sum_{i=1}^{p}\frac{||\hat{x}_i - x_i||_2}{d_k} \tag{1.1}$$

где K – количество изображений в датасете, p – количество ключевых точек лица, x_i – координаты i-ой ключевой точки, $\hat{x}_i$ – предсказанные координаты i-ой ключевой точки, d_k – нормировочный коэффициент. Нормировочный коэффициент может быть разным для разных датасетов, но в большинстве случаев это расстояние между зрачками (inter-pupil), либо расстояние между внешними краями контура глаз (inter-ocular).

Существует и другие, менее принятые метрики. Например, FR (Failure Rate) и CED-AUC (Cumulative Error Distribution – Area Under Curve). Более подробно о метриках можно почитать, например, здесь [1].

1.3 Обзор датасетов для задачи вычисления ключевых точек лица

Существует несколько открытых датасетов, доступных для обучения и оценки качества алгоритмов. Каждый из датасетов включает изображение человека и соответствующую разметку ключевых точек лица. Разметка представлена в отдельном файле. Датасеты могут включать изображения следующих видов:

1. В контролируемой среде (например, в студии) или «в дикой природе»;
2. С различными условиями съемки человека, такими как наличие разных предметов на лице (солнцезащитные очки, маска, шарф), крупная поза, макияж и т.д.;
3. Реальные изображения или синтетические (лица генерируются с помощью некоторого алгоритма);
4. 2D или 3D ключевые точки лица.

Рассмотрим теперь самые часто используемые датасеты для задачи регрессии ключевых точек лица:

1. 300 Faces in-the-Wild (300W).

   Датасет состоит из нескольких других датасетов: LFPW, AFW, HELEN, XM2VTS и IBUG. Он содержит изображения, размеченные 68 ключевыми точками лица. При этом изображения в 300W сделаны при разных условиях съемки: разное освещение, цветовая гамма, эмоции, углы поворота лица. Часть датасета, используемая для тестирования, разбита на 3 группы сложности. Для сравнений результатов по данному датасету зачастую приводят метрику NME, используя inter-pupil либо inter-ocular нормализацию.

2. Annotated Facial Landmarks in-the-Wild (AFLW).

   AFLW содержит большее количество изображения, по сравнению с 300W, однако изображения размечены всего 21 ключевыми точками (хотя существует и версия с 68 точками). При этом датасет содержит много изображений с большим углом поворота лица. Авторы предлагают делить датасет на 2 части: AFLW-Frontal (с изображениями, где лицо почти полностью смотрит в камеру) и AFLW-Full (все изображения). Для сравнений результатов по данному датасету используется метрика NME, нормализованная на длину диагонали окаймляющего лица прямоугольника.

3. Caltech Occluded Faces in-the-Wild (COFW).

   Датасет содержит преимущественно изображения лиц, частично закрытых либо сторонними предметами, такими как, например, шарфом, либо частями человеческого тела, например, рукой. Лица размечены 21 точкой. Помимо NME метрики, всё также использующую inter-pupil или inter-ocular нормализацию, здесь для сравнений результатов иногда используется Failure Rate.

4. Wider Facial Landmarks in-the-Wild (WFLW).

   Лица в этом датасете размечены 98 точками. И в сравнении с предыдущими датасетами, WFLW содержит больше изображений, сделанных в необычных условиях, например, изображения лиц с макияжем, широким спектром эмоций. Все перечисленные ранее метрики используется для сравнения в данном датасете: NME (inter-ocular нормализация), Failure Rate, CED-AUC. Также для каждого изображения в датасете содержится описания условий съемки. Данная информация тоже может быть использована в датасете.

В данной же работе будет использоваться более новый датасет Landmark guided face Parsing dataset (LaPa), созданный в попытке повысить точность и качество обучения. Подробное описание данного датасета будет дано в разделе 3.1.

ГЛАВА 2
ОБЗОР МЕТОДОВ РЕГРЕССИИ КЛЮЧЕВЫХ ТОЧЕК ЛИЦА

2.1 Общие сведения об алгоритмах регрессии ключевых точек лица

Регрессия ключевых точек лица на изображении является достаточно сложной задачей из-за как жесткой (масштабирование, поворот и параллельный перенос), так и не жесткой (например, изменение выражения лица) деформации лица.

Условно, все алгоритмы для регрессии ключевых точек лица можно разделить на 3 большие группы:

1. Ранние алгоритмы, использующие статистические подходы в своей основе;
2. Алгоритмы, основанные на ансамблях решающих деревьев и градиентном бустинге;
3. Нейронные сети.

2.2 Ранние алгоритмы регрессии ключевых точек лица

Наиболее известными алгоритмами этой группы являются Constrained Local Model (CLM) и Active Appearance Model (AAM). Они обладают достаточно хорошей точностью предсказания для изображения в контролируемых условиях (при надлежащем освещении и фронтальном ракурсе лица), однако абсолютно непригодны для изображений in-the-wild.

AAM методы являются медленными, что не позволяет их эффективно использовать в приложениях реального времени. При этом state-of-the-art CLM методы занимают порядка 0,1с на процессоре 2.5 GHz Intel Core 2 Duo, что при определенных ограничениях позволяет их использовать в некоторых приложениях реального времени. Описание AAM методов может быть найдено в статье [2]. Опишем общую идею CLM методов, однако подробное описание и обзор может быть также найден в статье [2].

CLM методы имеют 2 составляющие: некоторая модель, которая находит координаты всех p ключевых точек лица, и p «локальных экспертов», каждый из которых оценивает, насколько точно модель предсказала координаты ключевой точки. «Локальные эксперты» также являются моделями,

которые учатся на некоторых маленьких кусочках картинки вокруг заданной ключевой точки. Комбинируя вместе описанные модели, CLM методы пытаются оптимизировать функцию, которую можно интерпретировать как вероятность, что ключевые точки лица были правильно классифицированы, с учетом допущения, что все «локальные эксперты» работают независимо.

2.3 Алгоритмы, основанные на ансамблях решающих деревьев и градиентном бустинге

Одним из самых эффективных алгоритмов данной группы является алгоритм, предложенный и описанный в статье [3]. Алгоритм позволяет добиться обработки одного изображения на центральном процессоре всего за 1ms, что позволяет уже использовать данные алгоритм в приложениях реального времени без всяких ограничений.

Одной из основных проблем регрессии ключевых точек лица является наличие своеобразной циклической зависимости: нам необходимы надежные признаки, чтобы точно предсказывать форму, и при этом нам нужна точная оценка формы для извлечения надежных признаков. Очень хорошо с решением данной проблемы справляются итеративные подходы, и в частности – ансамбль решающих деревьев. На каждой итерации будет уточняться текущая оценка формы и извлекаться признаки в соответствии с этой оценкой. В результате, признаки будут становиться все более надежными, а оценка формы все более точной.

Ансамбль решающих деревьев представляет собой набор из T сильных регрессоров r_t – решающих лесов, каждый из которых состоит из K слабых регрессоров g_k – решающих деревьев. Обучение ансамбля решающих деревьев производиться с помощью градиентного бустинга, используя MSE функцию потерь.

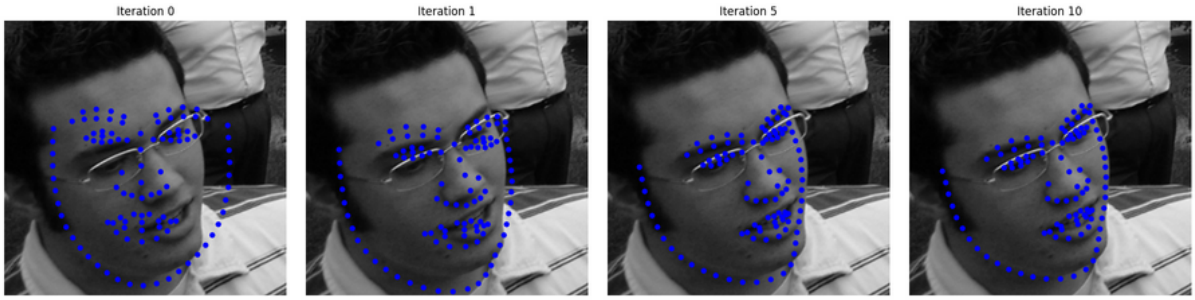

Рисунок 2.1 — Пример работы алгоритма. Уточнение ключевых точек происходит на каждой итерации.

Данный алгоритм реализован в Dlib – популярной библиотеке для машинного обучения с открытым исходным кодом, написанной на C++. Также

в открытом доступе можно найти предобученную на датасете 300W модель для Dlib.

Отметим также, что модель может стать очень легковесной, ввиду того, что решающие деревья с помощью квантизации значений листьев и алгоритма Хаффмана можно эффективно сжать, значительно уменьшив размеры модели, и при этом не очень драматически потеряв в точности.

2.4 Нейронные сети

Современные алгоритмы распознавания лиц in-the-wild основаны на нейронных сетях. Они делятся на 2 основные категории: методы прямой (или координатной) регрессии, когда модель предсказывает координаты x, y непосредственно для каждой ключевой точки, и методы регрессии на основе тепловой карты (heatmap), где для каждой ключевой точки строится специальная тепловая 2D-карта. Значения на тепловой карте могут быть интерпретированы как вероятности местоположения ключевой точки в определенной области картинки. Как правило, argmax или его модификация используется для получения точных координат ключевой точки из тепловой карты.

Методы прямой регрессии обычно используют широко известные модели, впервые построенные для ImageNet challenge, такие как ResNet, MobileNetV2, MobileNetV3, ShuffleNet-V2. Методы, основанные на тепловых картах, обычно используют архитектуру Hourglass, но также HRNet и CU-Net. Подробно про основные методы регрессии ключевых точек лица изложено в [1].

В данной работе мы сфокусируемся на быстрых методах регрессии ключевых точек, ввиду их большей значимости на практике. Это означает, что мы не будем касаться методов, основанных на тепловых картах, которые требуют гораздо больше вычислительных мощностей и времени и не применимы, например, в ситуациях обработки изображений в реальном времени. Тем не менее, методы, использующие тепловые карты, важны ввиду их высокой точности. В частности, они могут быть использованы для полуавтоматической разметки данных, чтобы людям не приходилось размечать новые данные полностью с нуля.

2.5 Сравнение

Ранние модели регрессии ключевых точек лица не имеют преимуществ перед нейронными сетями и ансамблями решающих деревьев и поэтому редко используются в прикладных целях.

Методы, основанные на ансамблях решающих деревьев и градиентном бустинге, являются очень быстрыми и легковесными. Скорости в 1ms, особенно на центральном процессоре, добиться с помощью даже самых легких нейронных сетей на данный момент не получается. Также, ввиду открытой

реализации в библиотеке DLib, метод до сих пор часто используется на практике.

Нейронные сети в последние годы продемонстрировали поразительное улучшение качества для задачи регрессии ключевых точек лица изображений in-the-wild. Однако необходимо понимать, что в первую очередь, нейронные сети были разработаны для выполнения на серверах с большим количеством графических процессоров. В последнее время также становится популярным применение нейронных сетей на мобильных устройствах с мощными графическими процессорами и нейронными процессорами (ИИ-ускорителями). Тем не менее, вычислительная сложность моделей, основанных на нейронных сетях, гораздо больше, что не всегда подходит. Их стремительный прогресс стал возможен благодаря развитию эффективности и производительности оборудования.

ГЛАВА 3
ПРИМЕНЕНИЕ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕГРЕССИИ 2D КЛЮЧЕВЫХ ТОЧЕК ЛИЦА

3.1 Обзор данных LaPa

LaPa (Landmark guided face Parsing) датасет содержит 22176 изображений разнообразных лиц, где для каждого изображения размечены 106 ключевых точек лица и создана специальная карта категорий, где каждая категория представляет собой отдельную часть лица. Всего таких категорий 11: волосы, кожа лица, левая/правая бровь, левый/правый глаз, нос, верхняя/нижняя губы, внутренняя часть рта и фон.

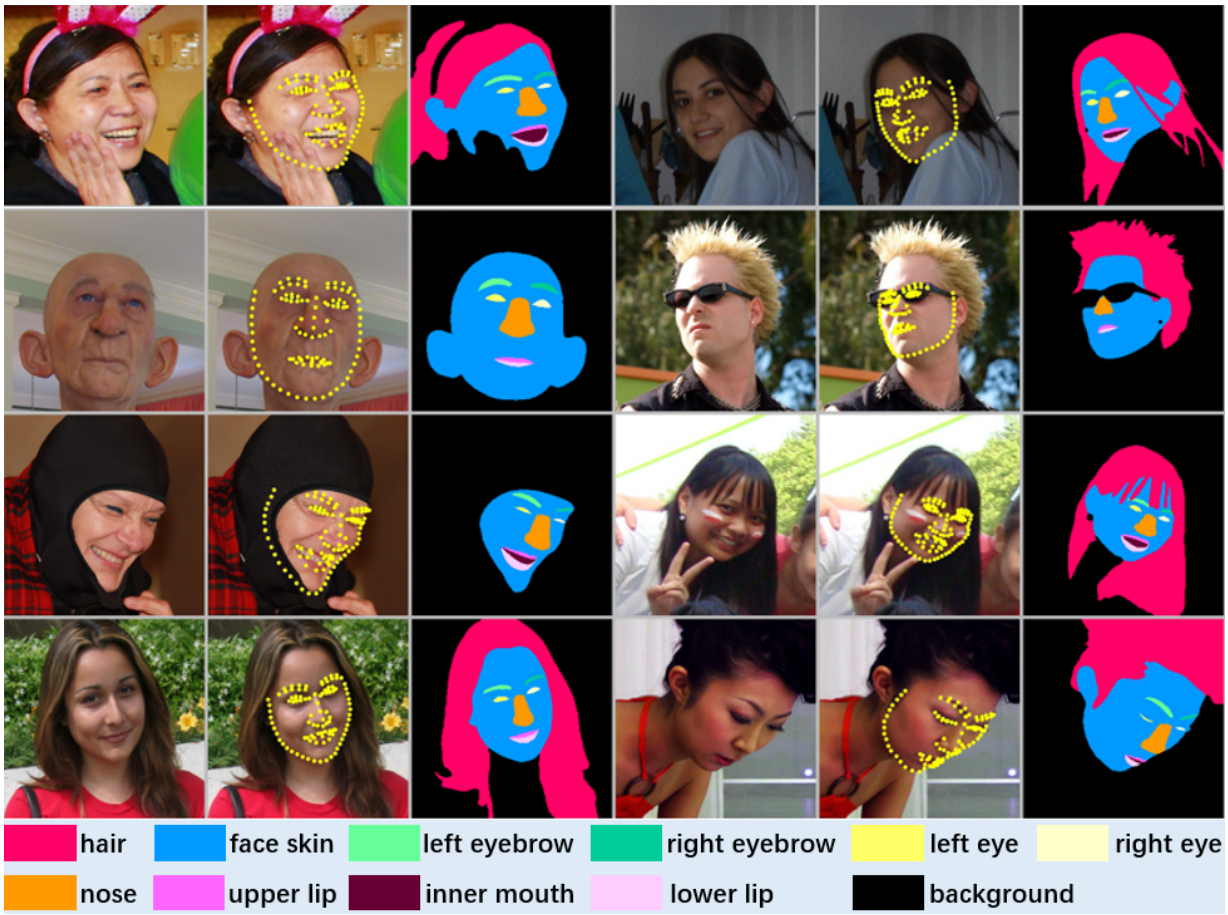

Рисунок 3.1 — Примеры размеченных ключевых точек лица и карт категорий для датасета LaPa

Большинство общедоступных датасетов размечены менее чем 100 ключевыми точками, что недостаточно для описания лица с хорошей детализацией. Например, датасет 300W описывает только верхнюю границу брови и делает это всего лишь 5 точками. Популярный датасет Helen содержит разметку

для 194 ключевых точек, но количество изображений составляет всего лишь 2330 штук, а также нет ключевых точек на переносице. Однако, как хорошо известно, хорошие данные для обучения имеют решающее значение для достижения хороших результатов, особенно с помощью методов глубокого обучения. Именно с целью повышения точности и качества обучения и был создан датасет LaPa.

Стоит также отметить, что для разметки ключевых точек на изображениях в датасете использовалась своя технология, которая позволила упростить финальный процесс разметки для людей и повысила качество этой разметки. Более подробно процесс описан в статье [4].

В обучении использовались заранее выделенные авторами части датасета:

1. Тренировочные данные, состоящие из 18176 изображений;

2. Валидационные данные, состоящие из 2000 изображений. Использовались для оценки качества обучения на каждой эпохе;

3. Тестовые данные, которые использовались для финального замера результатов. Объем также состовляет 2000 изображений.

3.2 Подготовка данных LaPa

В данной статье будут рассматриваться предобученные нейронные сети на задаче ImageNet. В связи с этим связаны стандартные для этого действия по предобработке изображений: перевод значений пикселей в диапозон $[0, 1]$, масштабирование изображения до размеров 224×224, нормализация изображений.

Ключевые точки лица после масштабирования изображения могут иметь координаты $[0, 224)$, если конечно лицо полностью помещается на изображение, что почти всегда верно. Нейронным сетям легче научиться предсказывать маленькие значения, которые к тому же симметричны относительно нуля. Поэтому для каждой ключевой точки P мы применим преобразование $(P - (112, 112))/112$, которое в большинстве случаев переведет значение координат точки P в диапазон $[-1, 1]$.

Важно также отметить, что на вход в нейросеть логично подавать не всю картинку, а только достаточно маленькую область, содержащую лицо, ключевые точки которого нужно найти. Для этого необходимо перед всеми остальными вышеописанными действиями выделить данную область на исходной картинке. Это можно сделать двумя способами:

1. Обрезать лицо по истинным ключевым точкам изображения.

2. Использовать некоторый уже обученный детектор лиц, для того чтобы получить ограничивающую лицо рамку.

Использование первого способа вместо второго привело бы к более лучшим результатам метрики на тестовых данных, однако более худшим результатам на данных вне нашего датасета. Это связано с тем, что при попытке применить нашу модель к какому-нибудь изображению вне нашего датасета, нам все равно пришлось бы получать ограничивающую лицо рамку с помощью детектора лиц. И зачастую детекторы лиц, особенно достаточно простые и быстрые, генерируют не самую качественную рамку. Если сверточная нейросеть всегда будет учиться на идеальных ограничивающих рамках, то увидев рамку, полученную от детектора, она не сможет выдать качественный результат, ибо свертки не инвариантны к разнообразным сдвигам. Данный недостаток первого способа можно с некоторой степенью нивелировать, если аугментировать ограничивающую рамку. Подробнее про это будет в разделе 3.3.

Так или иначе, первый способ является классическим примером того, как можно “подсматривать“ в целевые данные, однако его совместное со вторым способом использование на этапе обучения может помочь ускорить и улучшить качество обучения. В результате, в процессе обучения для получения ограничивающей лицо рамки мы будем случайно равновероятно выбирать первый или второй способ. На этапе валидации и тестирования мы будем использовать только второй способ. Влияние такого способа обучения показано в разделе 3.8.1. Полученную ограничивающую рамку мы будем увеличивать на 10% по каждой из осей. Это необходимо для того, чтобы не потерять некоторую часть лица при его детектировании либо при последующей аугментации ограничивающей рамки.

3.3 Аугментация данных

Аугментация данных является важным этапом, ибо она позволяет получить лучшие результаты обобщения модели, а также искусственно увеличить количество данных. Необходимость в аугментациях для достижения качественного результата будет явно показана в 3.8.1. Следующие способы аугментации изображений являются самыми популярными:

1. Отображение по вертикали или горизонтали.
2. Поворот на определенный угол.
3. Создание отступа.
4. Добавление шума.
5. Манипуляции с цветом.

В нашей задаче для обучения всех нейросетей будут использоваться следующие достаточно простые и тем не менее эффективные аугментации:

1. Случайный поворот изображения (на угол до 35°).

2. Манипуляции с цветом: изменение яркости, контрастности, насыщенности и оттенка изображения.

3. Случайный сдвиг границ ограничивающей лицо рамки.

Остановимся подробнее на последнем пункте. Аугментация в сущности представляет собой следующее: каждую из 4 границ мы будем равновероятно сдвигать на $[-5\%, +5\%]$ от текущего положения, где проценты считаются относительно размеров рамки по соответствующей оси. Данная аугментация в текущей задаче заменяет собой создание отступа. Она моделирует неточность, которая может возникнуть при определении лица с помощью детектора лиц.

3.4 Общий алгоритм обучения нейронной сети регрессии ключевых точек лица

Как уже было отмечено, в данной статье будут рассматриваться предобученные нейронные сети на задаче ImageNet. Задача регрессии ключевых точек лица достаточно сильно отличается от задачи ImageNet, тем не менее, использование fine-tuning для предобученных на ImageNet сетях в задачах, связанных с изображениями, является почти стандартом в мире обучения нейронных сетей. Для демонстрации данного утверждения в разделе 3.8.1 можно найти результаты обучения нейронной сети со случайно инициализированными весами.

Все рассмотренные ниже архитектуры будут использоваться лишь с одной необходимой модификацией: в самом последнем линейном слое количество выходных нейронов будет заменено на 212, что является удвоенным количеством ключевых точек лица в датасете LaPa.

Все веса нейронной сети будем оставлять "размороженными" и будем обновлять их в процессе обучения. Впрочем, можно подумать, что первые слои нейронной сети извлекают достаточно общие для всех изображения признаки, и попробовать "заморозить" первые слои. Результаты данного эксперимента можно найти в разделе 3.8.1.

В качестве оптимизатора будет использоваться Adam. Шаг обучения будет уменьшаться в 4 раза, если функция потерь на валидации не будет улучшаться спустя 4 эпохи. Однако первые 3 эпохи будет использоваться "разогрев" (warmup) – начальный шаг линейно будет увеличиваться от значения 10^{-5} до 10^{-3}. Данный прием помогает "привыкнуть" оптимизатору к данным и не сильно обновлять веса на первых этапах. Это нужно для того, чтобы не "сломать" хорошие веса предобученной нейронной сети. Влияние данной

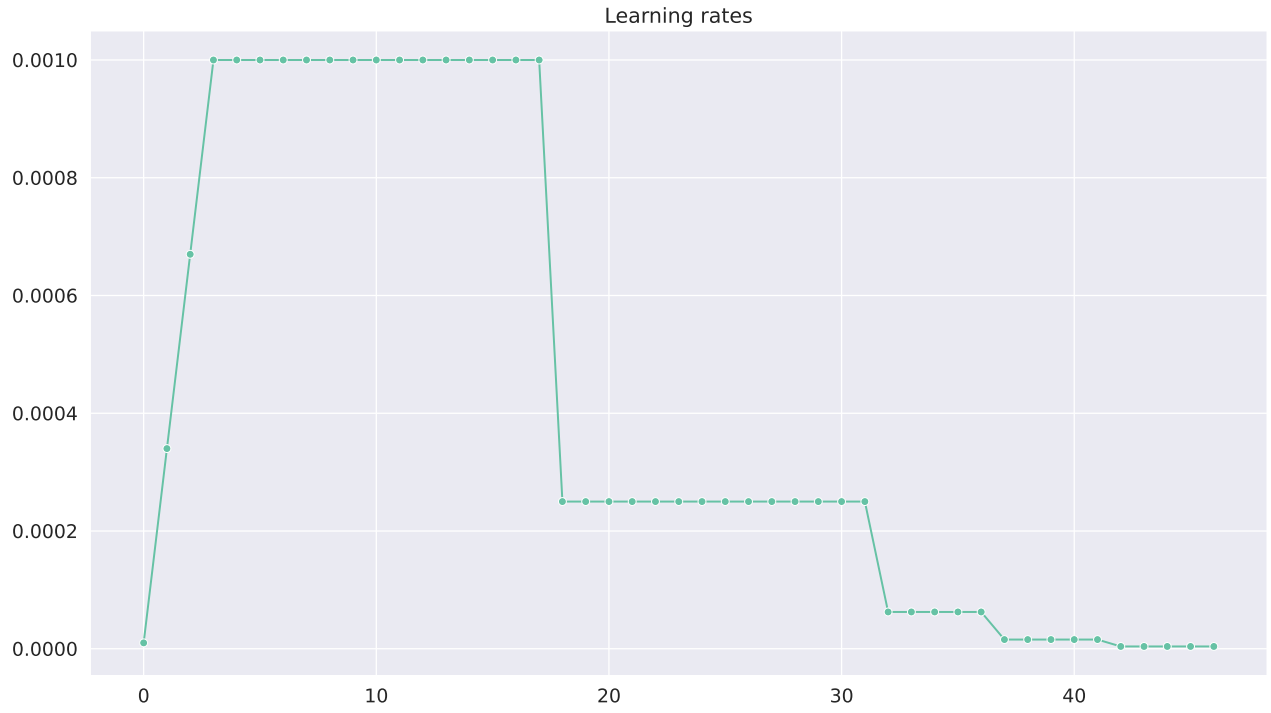

Рисунок 3.2 — Пример изменения шага обучения в зависимости от количества эпох

техники будет также исследовано в разделе 3.8.1. Пример изменения шага с ходом обучения приведен на графике 3.2:

В качестве функции потерь была выбрана стандартная в случае задач регрессии функция среднеквадратичной ошибки. Однако в силу некоторых особенностей, качество обучения можно сильно улучшить при использовании иной функции потерь. Подробнее об этом будет рассказано в разделе 3.8.2.

На каждой итерации мы будем подавать нейронной сети батч из 16 тренировочных картинок и делать шаг оптимизатора. Каждую эпоху будем замерять значения функции потерь и метрики NME, нормализованной на interpupil расстояние. В зависимости от значения функции потерь будет изменяться шаг обучения по описанному ранее алгоритму. При достижении значения шага 10^{-6} мы будем завершать обучение.

Обучения нейронных сетей будет происходить на языке Python с помощью библиотеки PyTorch. В связи с этим возникают некоторые особенности, которые будут описаны в 3.7. Весь исходный код обучения и применения нейронных сетей можно найти по ссылке `https://github.com/ArseniyTy/FaceLandmarkDetection`.

3.5 Регрессия ключевых точек лица на основе классических нейронных сетей

В этом разделе мы рассмотрим классические сверточные архитектуры, такие как ResNet и MobileNetV2. По каждой из архитектур будет кратко описаны основные революционные идеи. Также будут приведены результаты обучения данных нейронных сетей для рассматриваемой задачи.

3.5.1 Регрессия ключевых точек лица на основе ResNet18

Основная проблема, связанная с глубокими нейронными сетями – затухание градиента. В итоге, на первых слоях нейронной сети градиенты будут почти нулевыми и такие веса будут обновляться очень медленно. Для решения данных вопросов были придуманы функция активации ReLU, слои батч нормализации. Все это используется в архитектуре ResNet, однако авторы статьи [5] предложили еще один революционный способ борьбы с затуханием градиентов: использование skip connections. Также после данной статьи стал повсеместно использоваться Global Average Pooling, чтобы получить вектор признаков из набора карт-признаков, полученных после применения сверток. Данный подход дал возможность использовать нейронную сеть для разных размеров входных изображений.

Мы рассмотрим версию архитектуры с 18 слоями – ResNet18. Она является менее точной, чем, например, ResNet152, однако является более легковесной с точки зрения количества параметров, а также более производительной с точки зрения времени применения.

На графиках 3.3 представлены значения функции потерь и NME метрики во время обучения.

В таблице 3.1 представлена дополнительная информация о результатах обучения и архитектуре ResNet18: итоговая метрика NME на тестовой выборке, количество эпох обучения, время инференса на ЦПУ и ГПУ, количество операций с плавающей точкой при применении нейронной сети (GFLOPS), количество параметров в нейронной сети.

NME	Эпох	ЦПУ инференс, мс	ГПУ инференс, мс	GFLOPS	Параметров
$0,0784$	47	53 ± 8	$4,0 \pm 0,6$	$1,81$	$11,7$M

Таблица 3.1 — Сводная информация по ResNet18

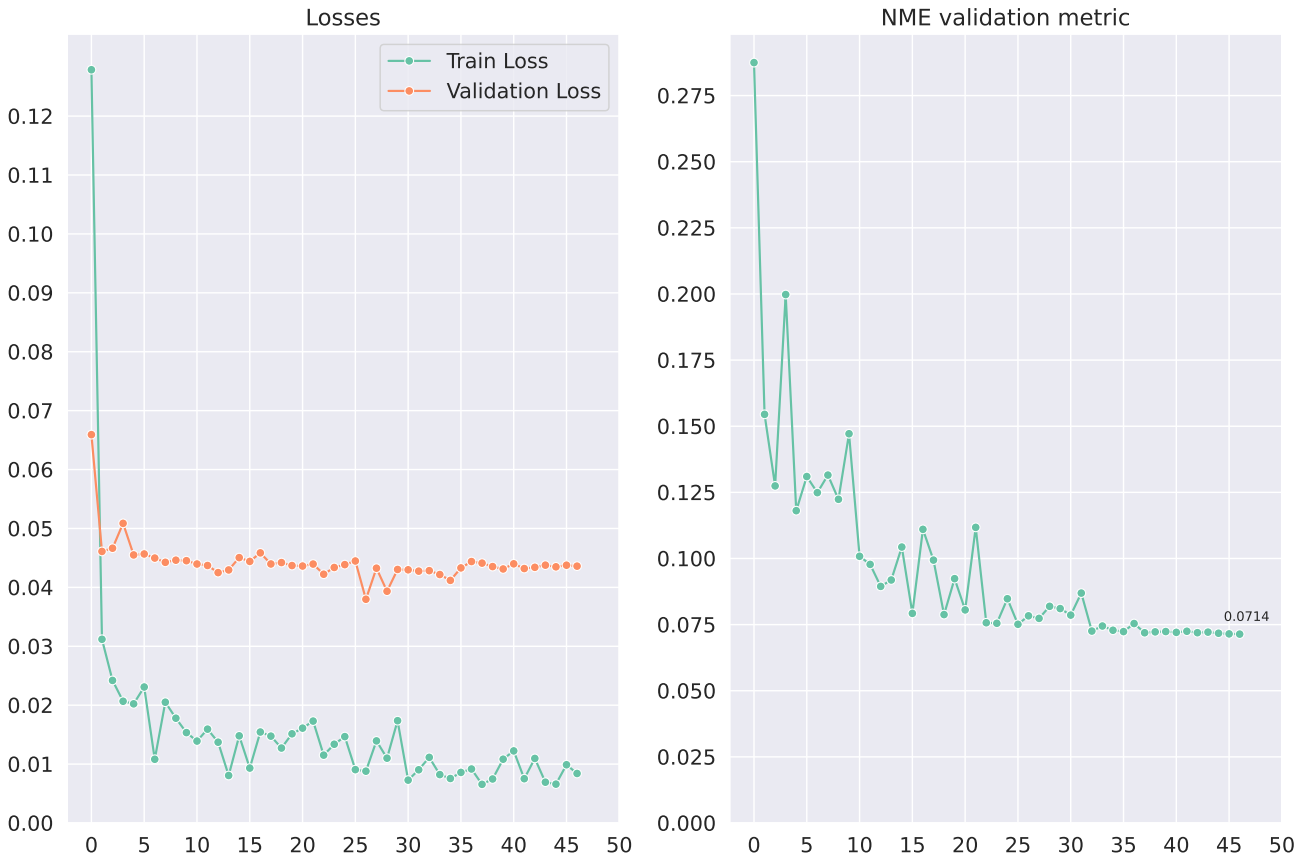

Рисунок 3.3 — Значения функции потерь и NME метрики для ResNet18

3.5.2 Регрессия ключевых точек лица на основе MobileNetV2

Мотивация создания архитектуры нейронной сети MobileNetV2, описанной в статье [6], заключается в создании легковесной и эффективной модели для мобильных устройств. Поэтому главной ее особенностью является низкое количество параметров вместе с быстрым инференсом. Это достигается главным образом за счет применения Depthwise Separable сверток, которые представляют обычную свертку как последовательное применение Depthwise свертки (1D свертки для каждого входного канала) и Pointwise свертки (2D свертки размера 1×1). MobileNetV2 использует Depthwise Separable свертки размером 3×3, которые требуют в 8-9 раз меньше вычислений, чем стандартные свертки, при лишь небольшом снижении точности.

Каждый блок MobileNetV2 сначала отображает входной тензор в пространство большей размерности (большее число каналов). Авторы статьи выдвигают гипотезу, которая эмпирически подтверждается, что полученное блоком на первом этапе пространство большой размерности может быть отображено в пространство меньшей размерности без потери полезной информации. Это также означает, что одна и та же информация содержится сразу в нескольких каналах, что позволяет применить функцию активации ReLU, которая вносит требуемую нелинейность в нейронную сеть, и при этом не по-

терять полученную информацию. Блок заканчивается 1×1 сверткой с линейной функцией активации, понижающей число каналов. Данный слой в статье был назван linear bottleneck. MobileNetV2 также использует skip connections между bottleneck для лучшего прохождения градиента.

На графиках 3.4 представлены значения функции потерь и NME метрики во время обучения.

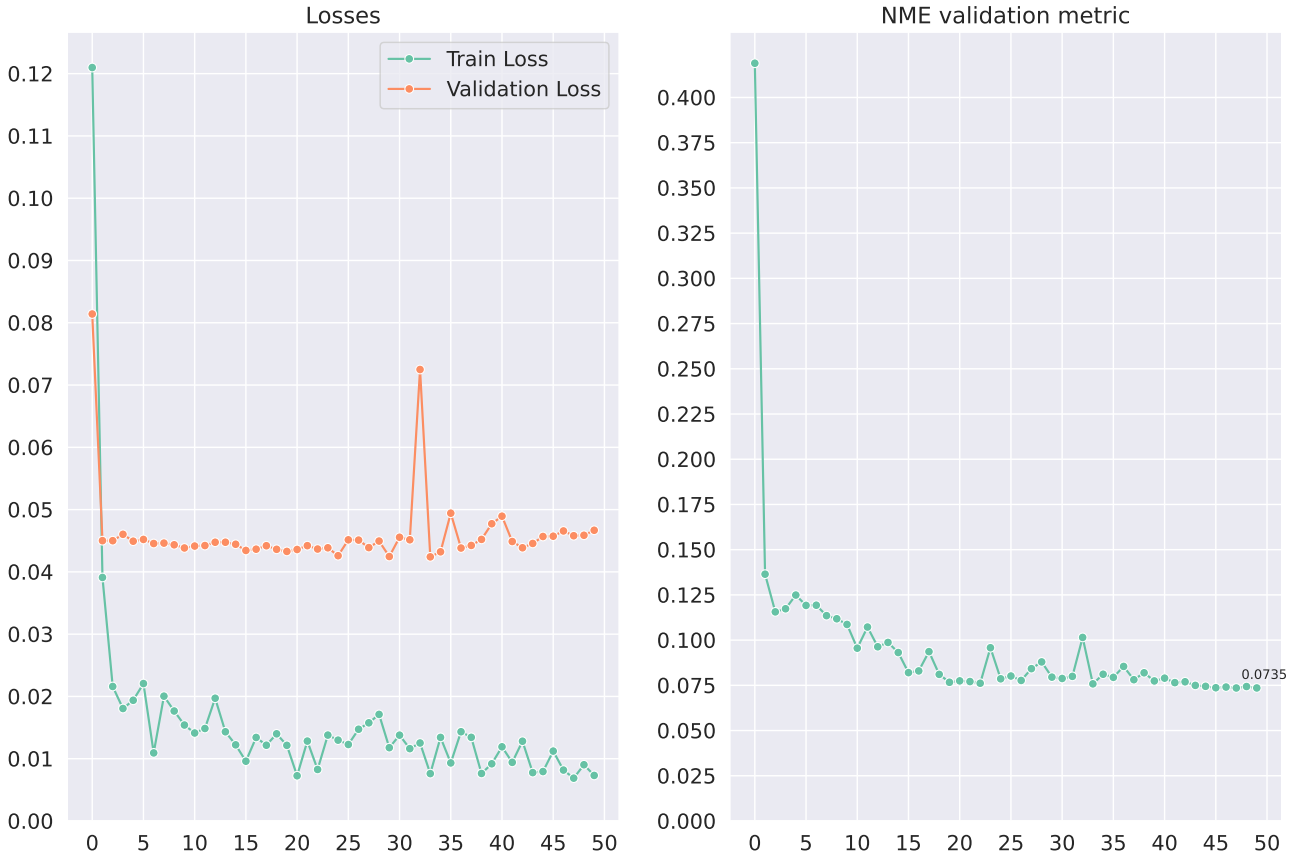

Рисунок 3.4 — Значения функции потерь и NME метрики для MobileNetV2

В таблице 3.2 представлена дополнительная информация о результатах обучения и архитектуре MobileNetV2.

NME	Эпох	ЦПУ инференс, мс	ГПУ инференс, мс	GFLOPS	Параметров
$0,0773$	50	29 ± 3	$7,4 \pm 1,8$	$0,3$	$3,5$M

Таблица 3.2 — Сводная информация по MobileNetV2

3.6 Регрессия ключевых точек лица на основе современных топологий нейронных сетей

В этом разделе мы рассмотрим более современные архитектуры нейронных сетей, такие как EfficientNet, MobileNetV3, MobileViT. По каждой из

архитектур будут описаны их основные идеи. Также будут приведены результаты обучения данных нейронных сетей для рассматриваемой задачи.

3.6.1 Регрессия ключевых точек лица на основе EfficientNetB0

Архитектура EfficientNet является одной из первых архитектур, полученных при помощи NAS (Neural Architecture Search), которая взяла SOTA на задаче ImageNet. После этого, почти все архитектуры перестали подбираться вручную. Для поиска архитектуры пространство параметров архитектуры было выстроено в иерархичную структуру. Поиск производился с помощью обучения с подкреплением, используя в качестве награды специальную функцию от точности модели и количества операций с плавающей точкой.

В качестве функции активации вместо ReLU стала использоваться SiLU (Swish), где $SiLU(x) = x\sigma(x)$, где $\sigma(x) = \frac{1}{1+e^{-x}}$. В более ранних работах было показано, что данная функция активации обычно позволяет добиться более высоких результатов.

В архитектуре также стал использоваться Squeeze-and-Excitation блок (SE-блок). Сверточная нейросеть учитывает разные входные каналы одинаково. Данный же блок с помощью Global Average Pooling, нескольких полносвязанных слоев и функции активации $\sigma(x)$ позволяет придать обучаемый вес каждому из каналов. SE-блок может быть добавлен в почти любую сверточную архитектуру, при этом он легковесный и требует очень малое количество дополнительных вычислений. Авторы статьи [7] показывают, что, добавляя SE-блоки в ResNet50, мы можем ожидать почти такой же точности, какую обеспечивает ResNet101.

Главной же особенностью статьи [8] являлось исследование масштабирования сверточных нейронных сетей. В статье было сделано 2 важных наблюдения:

1. Увеличение любого из параметров: ширины нейросети, её глубины или разрешения входного изображения, повышает точность, но для моделей большого размера прирост точности значительно уменьшается.

2. Для достижения большей точности и эффективности крайне важно сбалансировать все 3 параметра во время масштабирования сверточной сети.

Исходя из этих двух наблюдений авторы предложили свой подход для масштабирования произвольной модели. С помощью NAS была найдена самая базовая версия модели EfficientNetB0, а затем с помощью предложенного алгоритма были получены еще 6 увеличенных версий модели EfficientNetB1 - EfficientNetB7, каждая из которых в своё время являлась лучшей по точности при том же количестве параметров и времени инференса. В данной работе

мы рассмотрим лишь EfficientNetB0, как более легковесную и быструю архитектуру.

На графиках 3.5 представлены значения функции потерь и NME метрики во время обучения.

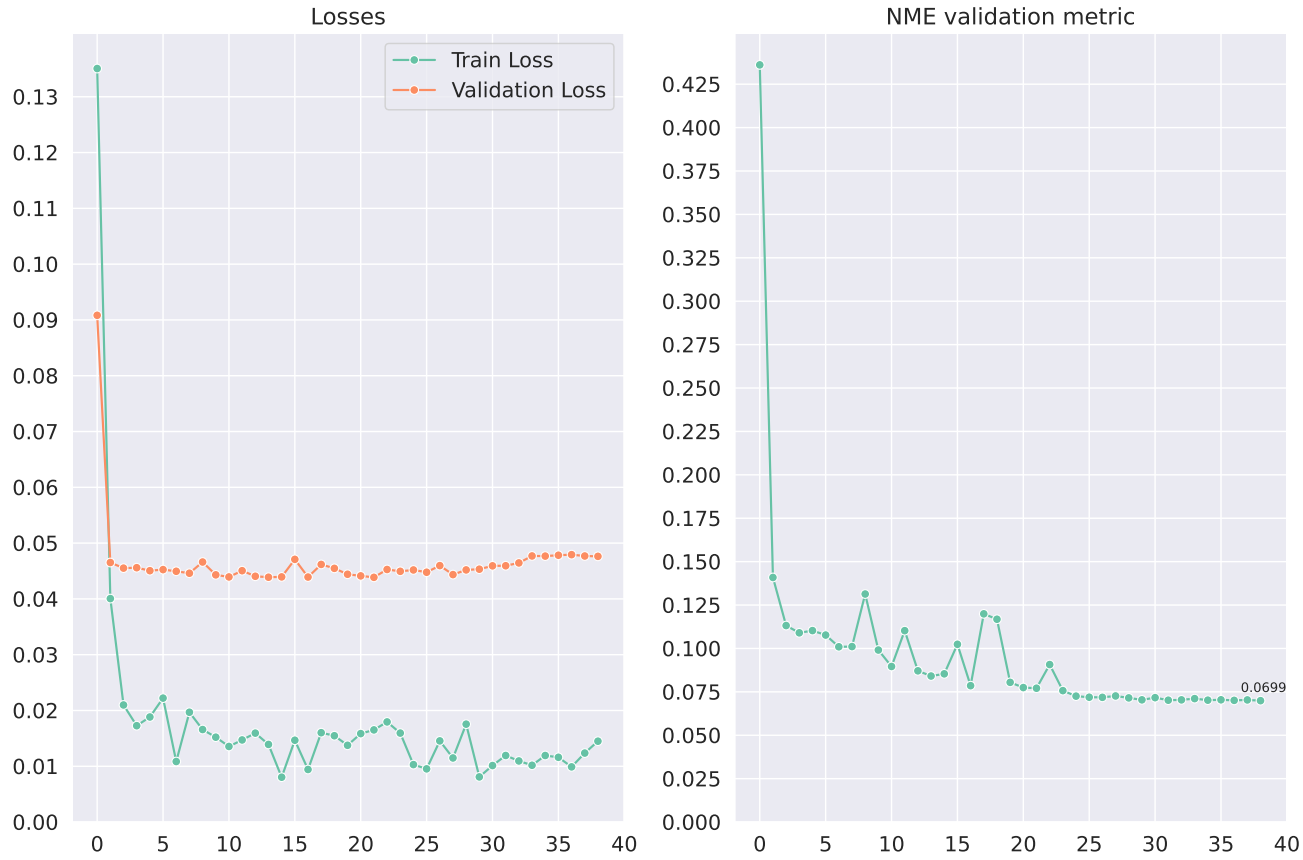

Рисунок 3.5 — Значения функции потерь и NME метрики для EfficientNetB0

В таблице 3.3 представлена дополнительная информация о результатах обучения и архитектуре EfficientNetB0.

NME	Эпох	ЦПУ инференс, мс	ГПУ инференс, мс	GFLOPS	Параметров
$0,0737$	39	41 ± 5	$11,3 \pm 2,0$	$0,39$	$5,3$M

Таблица 3.3 — Сводная информация по EfficientNetB0

3.6.2 Регрессия ключевых точек лица на основе MobileNetV3

Для поиска архитектуры MobileNetV3 авторы статьи [9] также используют NAS. Также используется алгоритм NetAdapt для поиска количества фильтров для каждого слоя. В третьей версии архитектуры были изменены

дорогостоящие слои в начале и конце сети. Эти изменения позволяли сэкономить 9 миллисекунд времени инференса. В архитектуру также добавились SE-блоки и Swish в качестве функции активации. Однако авторы статьи заметили две важные особенности Swish:

1. Вычислении функции $\sigma(x)$ является дорогостоящей операцией, особенно на мобильных устройствах. Вместо этого было предложено заменить функцию $\sigma(x)$ на ее линейный грубый аналог: $hard\text{-}\sigma(x) = \frac{ReLU6(x+3)}{6}$, где $ReLU6(x) = min(6, max(0, x))$. В итоге получаем функцию активации $hard\text{-}swish = x\frac{ReLU6(x+3)}{6}$. Данный аналог почти не снижает итоговую точность модели. В SE-блоках также используется $hard\text{-}\sigma(x)$.

2. Функция активации swish (hard-swish) начинает играть важную роль на более глубоких слоях нейронной сети. Поэтому hard-swish используется только во второй половине модели, а в первой половине используется классический ReLU.

Авторы статьи предложили 2 версии архитектуры: MobileNetV3-Large и MobileNetV3-Small. Они отличаются количеством параметров и временем инференса. MobileNetV3-Large показала на 3.2% более высокую точность на задаче ImageNet, при этом сократив время инференса на 20%, по сравнению с MobileNetV2. Далее в экспериментах мы будем использовать версию MobileNetV3-Large.

На графиках 3.6 представлены значения функции потерь и NME метрики во время обучения.

В таблице 3.4 представлена дополнительная информация о результатах обучения и архитектуре MobileNetV3Large.

NME	Эпох	ЦПУ инференс, мс	ГПУ инференс, мс	GFLOPS	Параметров
$0,0788$	32	26 ± 3	$8,5 \pm 1,1$	$0,22$	$5,5$M

Таблица 3.4 — Сводная информация по MobileNetV3Large

3.6.3 Регрессия ключевых точек лица на основе MobileViT

Было показано, что при огромном количестве разнообразных аугментаций, интенсивной L2-регуляризации и дистилляции ViTs (Vision Transformers) можно обучить на наборе данных ImageNet и достигнуть производительности на уровне сверточных нейронных сетей. Такие трансформеры содержат огромное количество параметров и являются сложными в обучении. Добавление пространственной информации, а также использование вспомогательных

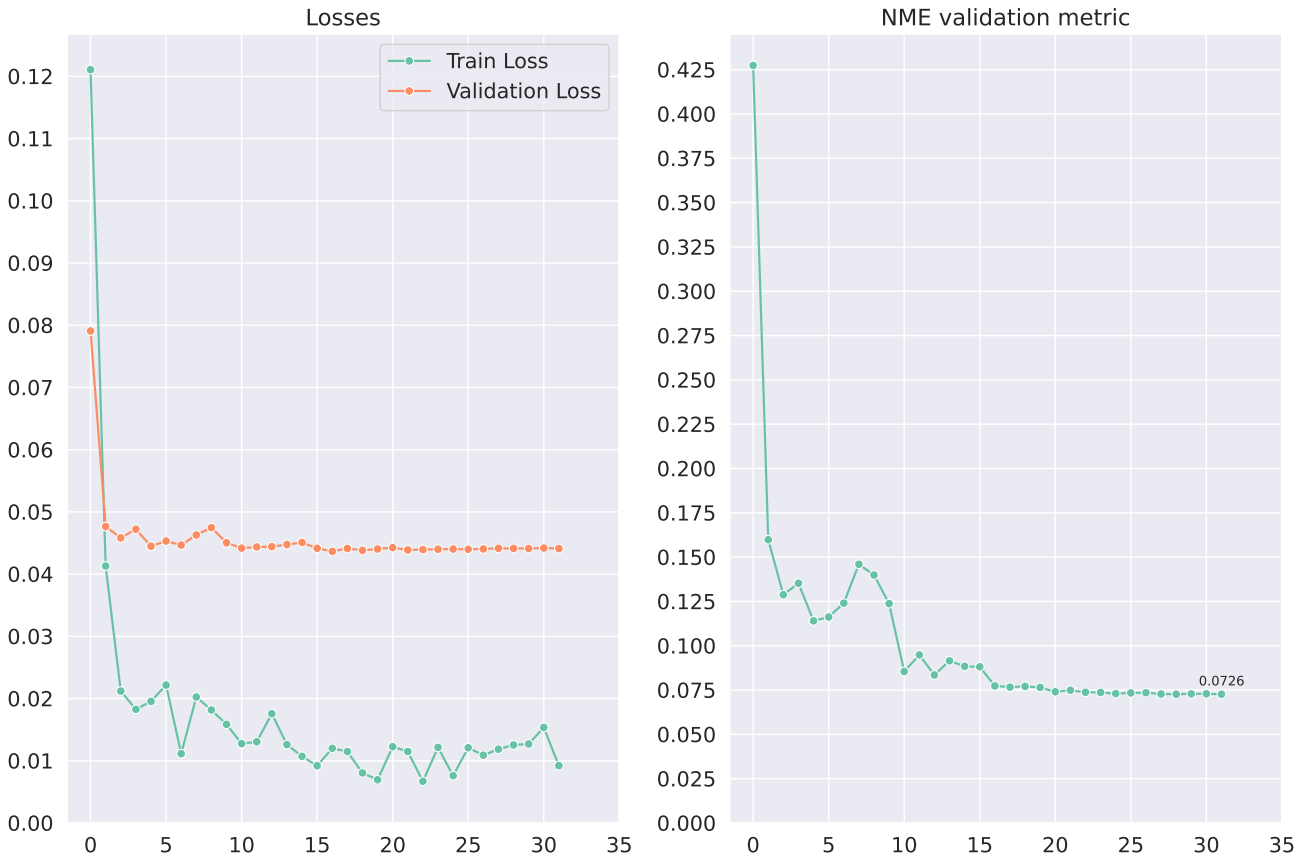

Рисунок 3.6 — Значения функции потерь и NME метрики для MobileNetV3Large

сверток позволяет достигнуть более хорошего качества и стабильности обучения, при этом облегчая процедуру аугментаций и регуляризации весов.

Авторы статьи [10] предлагают архитектуру MobileViT, которая эффективно совмещает концепции свёрточной нейронной сети и трансформеров. Ключевым в архитектуре является MobileViT блок, схематично изображенный на рисунке 3.7.

Его применение ко входному тензору можно условно разделить на 3 части:

1. Для начала ко входному тензору применяется стандартная свертка 3×3 для получения локальной пространственной информации, а затем Pointwise свертка для проецирования тензора в пространство большей размерности.

2. После этого тензор разбивается на неперекрывающиеся равные по размерам патчи. Каждый патч представляет собой некоторую прямоугольную часть картинки, где в каждом канале прямоугольник был вытянут в одномерный вектор. Далее следует серия трансформеров, которая обучается находить взаимосвязи между патчами, т.е. извлекает глобальные признаки. Полученные на выходе патчи мы собираем обратно в тензор.

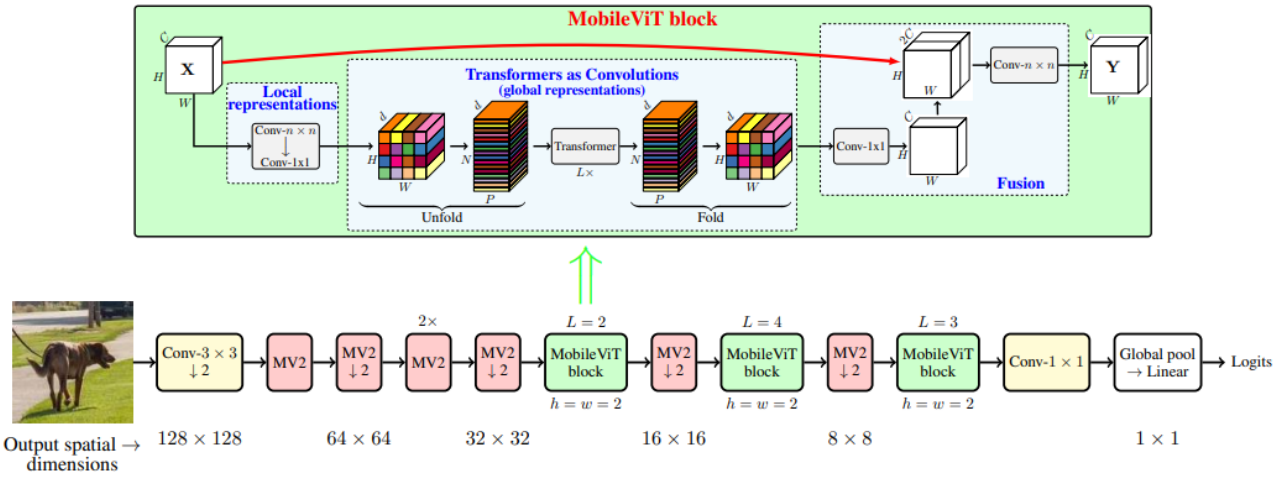

Рисунок 3.7 — Архитектура MobileViT и её ключевой элемент – MobileViT блок. Изображение взято из оригинальной статьи [10]

3. Применяем Pointwise свертки для проецирования тензора в пространство исходной размерности. Конкатенируем результат вместе с входным в блок тензором и применяем 3×3 свертку для получения финального результата. Конкатенация здесь играет роль skip connection и помогает градиентам не затухать.

В итоге, т.к. первая часть извлекает локальную информацию про некоторую часть входного изображения, а вторая часть извлекает глобальную связь между всеми патчами, то рецептивное поле MobileViT блока покрывает всё изображение.

Авторы статьи также предлагают разные по количеству параметров архитектуры: MobileViT-S, MobileViT-XS и MobileViT-XXS. В данной работе мы рассмотрим архитектуру MobileViT-S из $5,6$ миллионов параметров.

На графиках 3.8 представлены значения функции потерь и NME метрики во время обучения.

В таблице 3.5 представлена дополнительная информация о результатах обучения и архитектуре MobileViT.

NME	Эпох	ЦПУ инференс, мс	ГПУ инференс, мс	GFLOPS	Параметров
$0,0775$	36	80 ± 8	$12,4 \pm 1,7$	-	$5,6$M

Таблица 3.5 — Сводная информация по MobileViT

Отметим, что авторы статьи [10] приводят значение GFLOPS только для архитектуры MobileViT-XS, которая более чем в 2 раза меньше используемой нами версии, но для которой GFLOPS уже составляет $0,7$.

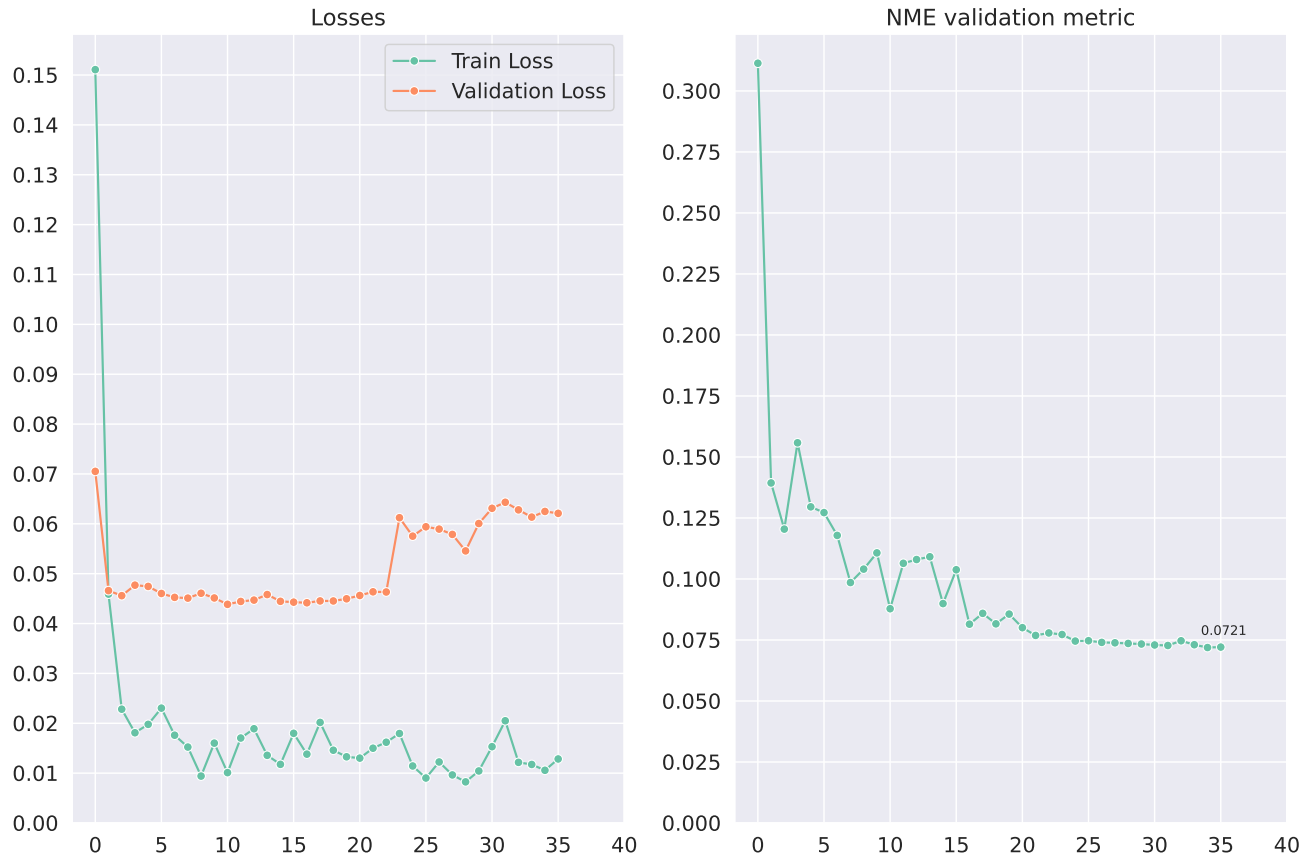

Рисунок 3.8 — Значения функции потерь и NME метрики для MobileViT

3.7 Анализ и сравнение топологий нейронных сетей для решении задачи регрессии ключевых точек лица

На всех графиках обучения различных архитектур мы видели похожую картину: значение функции потерь сильно падает на первых эпохах и далее слабо или почти не изменяется, особенно функция потерь на валидационной части данных. Тем не менее, значение метрики продолжает улучшаться и в конце обучения также выходит на константную прямую. Такое поведение кривой функции потерь, вообще говоря, может объясняться многими факторами. Например, данных для обучения может не хватать, а аугментации не слишком сильно разноображивают обучающую выборку. Возможно также более внимательной настройки требует шаг обучения и его изменение. Еще одним немаловажным фактором может служить тип функции потерь. В данном случае мы используем среднеквадратичную ошибку, которая, как известно, не является устойчивой к выбросам. Вполне вероятно, что из-за сильного влияния таких выбросов нейронная сеть не может в достаточно хорошей мере выучить нужные закономерности в данных. Подтверждение данной гипотезы можно найти в разделе 3.8.2.

Итоговые результаты метрики NME на тестовой выборке в зависимости

от различных параметров отображены на рисунке 3.9.

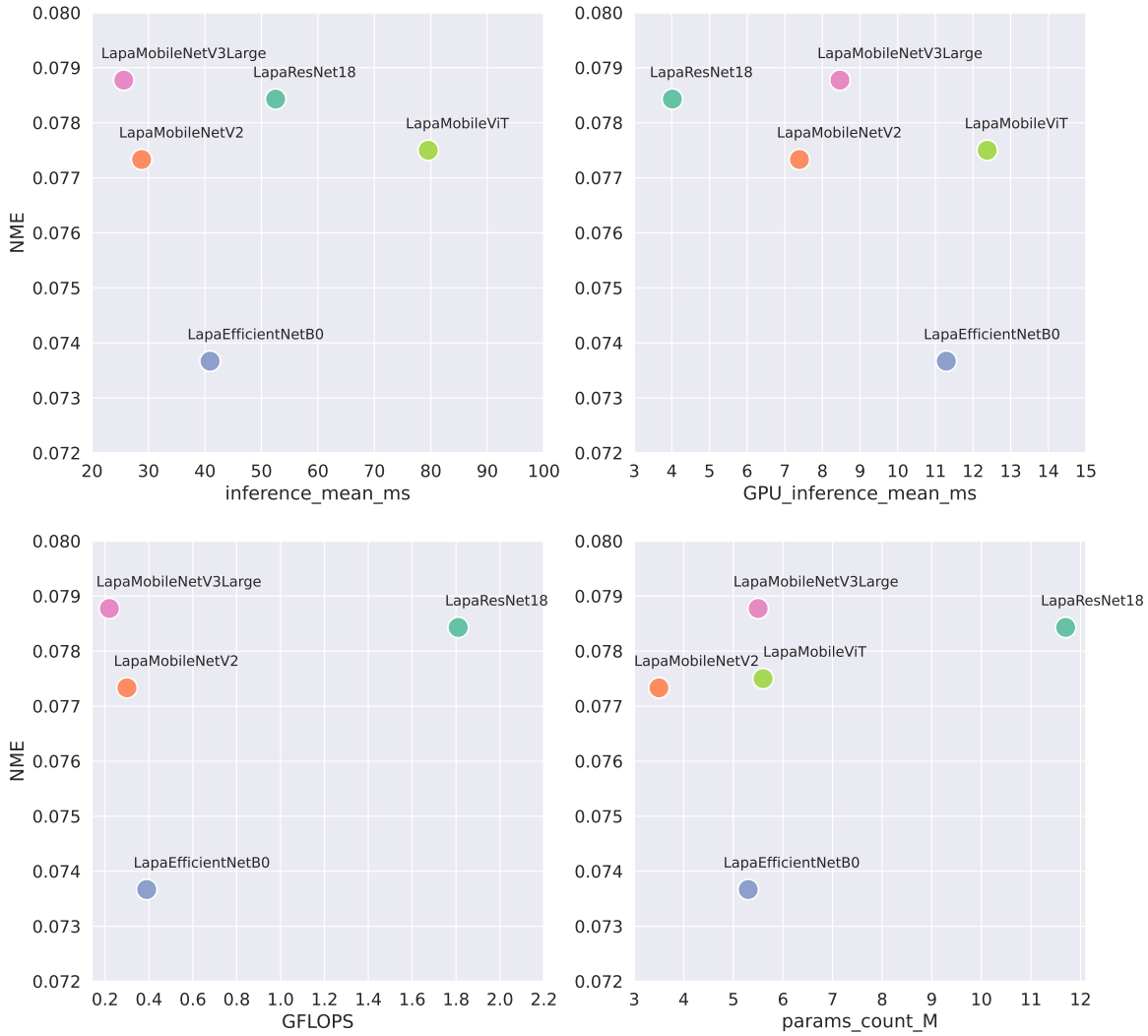

Рисунок 3.9 — Зависимость NME на тестовой выборке от различных параметров: времени инференса на ЦПУ, на ГПУ, GFLOPS, количества обучаемых параметров

На графиках наблюдается неоднозначная картина. Во первых, все архитектуры, кроме EfficientNetB0 дали примерно одинаковый показатель метрики. Однако очень важно, особенно в практических целях, насколько нейронные сети быстро решают задачу. Существуют разные показатели эффективности. Например, FLOPS – количество операций с плавающей точкой, которое совершает нейросеть. Однако FLOPS не учитывает такие инференс-факторы, как быстрота доступа к памяти, степень параллелизма и характеристики платформы. Нами было замерены также время инференса на ЦПУ модели Intel(R) Xeon(R) CPU @ 2.20GHz и время инференса на ГПУ модели Tesla P100. На графиках видно, что, например, ResNet18 имеет более чем в 8 раз больше FLOPS, чем MobileNetV3, однако всего в 2 раза хуже время инференса на ЦПУ и даже в 2 раза лучше время инференса на ГПУ. Такое

противоречивое время инференса ResNet18 относительно других архитектур объясняется тем, что все остальные рассмотренные архитектуры активно используют Depthwise свертки, которые в PyTorch реализованы крайне неэффективно. Также все остальные архитектуры больше оптимизированы для применения на мобильных устройствах.

MobileViT ожидаемо медленнее остальных архитектур из-за наличия трансформерных компонент. Более того, на мобильных устройствах разрыв скорее всего еще больше увеличится, ибо свертки достаточно хорошо оптимизированы под мобильные устройства, чего не скажешь о трансформерах.

3.8 Влияние разогрева, предобучения, аугментаций и функции потерь при обучении нейронной сети

3.8.1 Влияние разогрева, предобучения, аугментаций

В данном разделе мы покажем важность каждого из элементов общего алгоритма обучения нейронной сети, описанного в разделе 3.4, а также важность подготовки данных, описанной в разделах 3.2 и 3.3. Выберем в качестве архитектуры MobileNetV3, которая обучается быстрее других рассмотренных архитектур, и рассмотрим некоторые модификации процесса обучения и подготовки данных:

1. Отсутствие разогрева.
2. Использование предобученной сети с замороженными первыми слоями (будем замораживать первые 3 блока из 16 имеющихся в MobileNetV3-Large).
3. Использование не предобученной сети, т.е. со случайно инициализированными весами. В этом случае разогрев также естественно применять не будем.
4. Использование только детектора для получения ограничивающей рамки на этапе обучения. Напомним, что в стандартном алгоритме ограничивающая рамка может быть иногда (случайно) получена также по истинным ключевым точкам.
5. Отсутствие аугментаций поворота и манипуляции с цветом.
6. Отсутствие аугментаций поворота, манипуляции с цветом, а также сдвига границ ограничивающей рамки.

Результаты обучения MobileNetV3 с перечисленными модификациями видны в таблице 3.6.

Тип модификации	NME	Эпох
Без модификаций	0.0788	32
Без разогрева	0.0807	39
Замороженные слои	0.0817	39
Без предобучения	0.0881	56
Только детектор для получения рамки	0.0884	40
Без аугментаций поворота и манипуляции с цветом	0.0823	40
Предыдущее + без сдвига рамки	0.0931	33

Таблица 3.6 — MobileNetV3 с различными модификациями

Видим, что все рассмотренные модификации в той или иной мере ухудшают итоговое качество модели. Это подтверждает важность различных рассмотренных аспектов обучения.

3.8.2 Влияние функции потерь

Ранее в качестве функции потерь была выбрана стандартная в случае задач регрессии функция среднеквадратичной ошибки. Как известно, она не является устойчивой к выбросам. Вполне вероятно, что из-за сильного влияния таких выбросов нейронная сеть не может в достаточно хорошей мере выучить нужные закономерности в данных. Качество обучения можно сильно улучшить при использовании функции потерь Wing, предложенная в статье [11]. Она имеет вид:

$$\text{wing}(x) = \begin{cases} w \ln\left(1 + \frac{|x|}{\epsilon}\right), & \text{if } |x| < w \\ |x| - C, & \text{otherwise} \end{cases}$$

где $C = w - \ln\left(1 + \frac{w}{\epsilon}\right)$, w, ϵ – гиперпараметры, которые были выбраны как в оригинальной статье ($w = 15$, $\epsilon = 3$). Визуальное сравнение функций потерь изображено на графике 3.10.

Функция потерь Wing менее чувствительна к выбросам и гораздо более чувствительно к средним и малым ошибкам, что улучшает обучение в целом.

Обучим архитектуру EfficientNetB0, как самую точную из рассмотренных ранее, и архитектуру MobileNetV3, как, в теории, самую быструю с точки зрения времени инференса. Результаты обучения приведены в таблице 3.7.

Архитектура	NME (при обучении с MSE)	NME (при обучении с Wing)	Эпох
MobileNetV3	0.0788	0.0536	79
EfficientNetB0	0.0737	0.0524	93

Таблица 3.7 — Результаты обучения с функцией потерь Wing

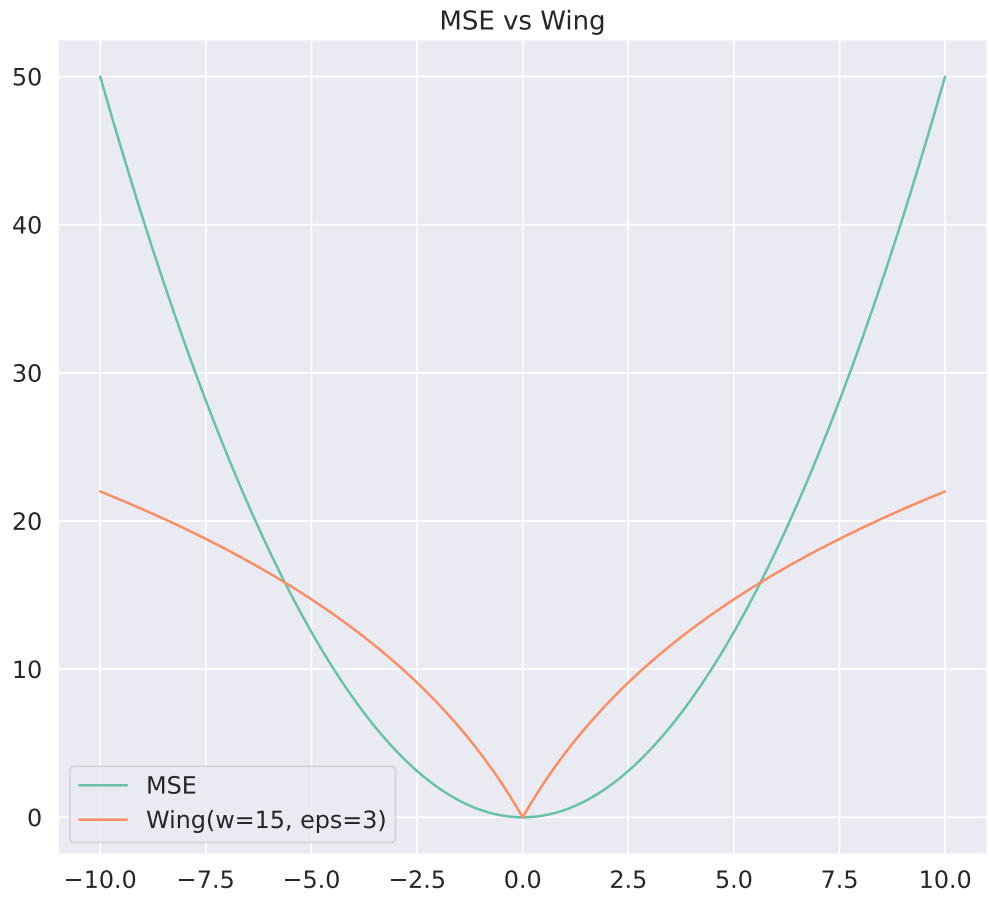

Рисунок 3.10 — Сравнение функций потерь среднеквадратичной ошибки и Wing

График обучения MobileNetV3 приведен на рисунке 3.11 (график обучения EfficientNetB0 имеет похожий вид).

Заменой лишь одной функции потерь нам удалось добиться поразительных улучшений в значении итоговой метрики. Также отметим, что в данном случае обучение шло гораздо дольше, а значение функции потерь Wing не выходило на плато так быстро, как это делала функция потерь среднеквадратичной ошибки.

Приведем также диаграмму, отображённую на рисунке 3.12, позволяющую увидеть, какой сильный прирост качества дает функция потерь Wing.

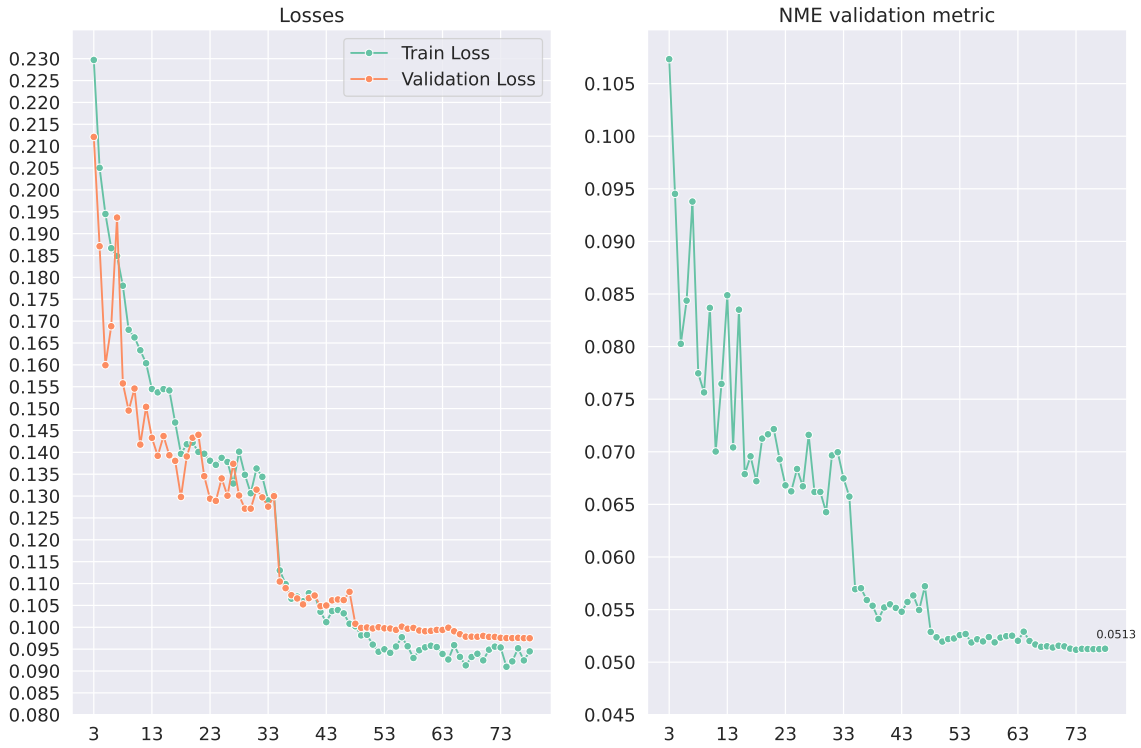

Рисунок 3.11 — Значения функции потерь Wing и NME метрики для MobileNetV3Large (начиная с 3 эпохи)

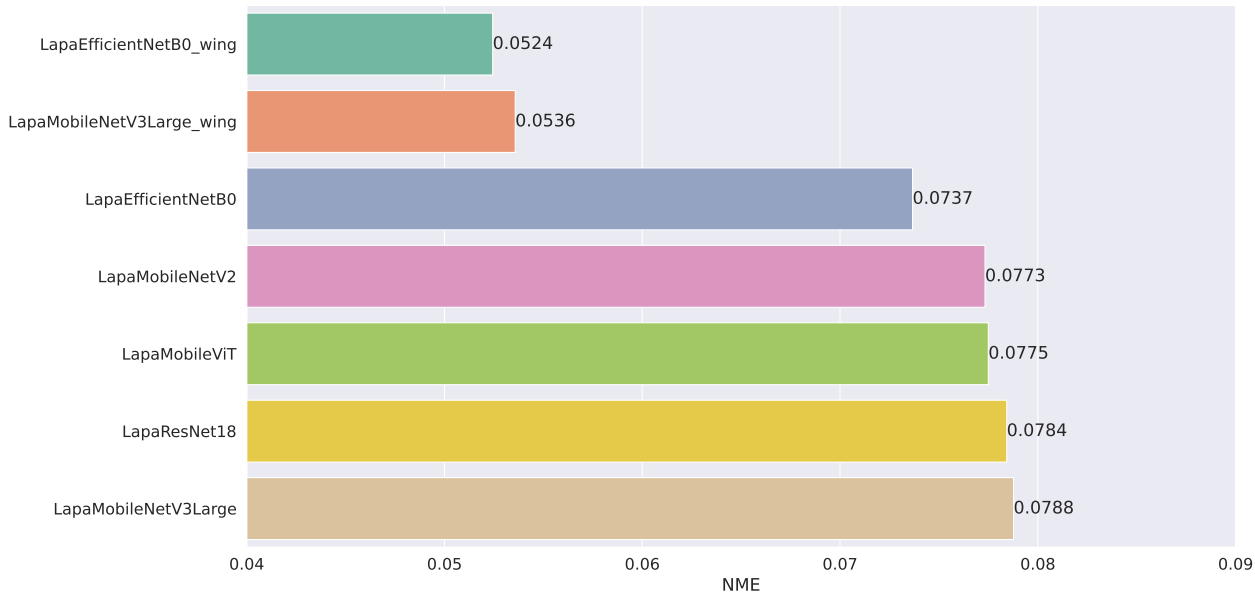

Рисунок 3.12 — Сравнение значений метрики NME для различных архитектур, обученных с помощью MSE и Wing функций потерь

ЗАКЛЮЧЕНИЕ

В ходе проекта:

1. Была рассмотрена постановка задачи нахождения ключевых точек лица, метрики оценки точности решений этой задачи, а также ее применение на прикладном уровне.

2. Сделан краткий обзор наиболее популярных датасетов для задачи регрессии ключевых точек лица.

3. Сделан обзор основных методов решения данной задачи, их преимуществ и недостатков.

4. Описана подготовка и аугментация данных датасета LaPa.

5. Подробно описан общий алгоритм обучения нейронной сети для решения задачи регрессии ключевых точек лица.

6. Описаны ключевые идеи различных классических и современных архитектур нейронных сетей: ResNet, MobileNetV2, EfficientNet, MobileNetV3, MobileViT.

7. Произведено обучение перечисленных выше нейросетей и исследована их точность. Также была исследована эффективность архитектур с точки зрения различных факторов: времени инференса на ЦПУ, на ГПУ, GFLOPS, количества обучаемых параметров.

8. Произведено исследование влияния различных аспектов в предложенном ранее алгоритме обучения: разогрева, предобученности архитектуры, количества обучаемых слоев нейросети, аугментаций, алгоритма получения ограничивающей лицо рамки.

9. Более подробно исследовано влияние функции потерь в общем алгоритме обучения нейронной сети. Выявлено, что функция потерь Wing позволяет достичь сильного прироста в качестве обучения.

Наиболее точную модель получилось создать на базе архитектуры EfficientNetB0. При обучении на MSE функцию потерь NME, нормализованная на расстояние между зрачками (inter-pupil), составила $0,0737$, а при обучении на функцию потерь Wing ошибка составила $0,0524$, что почти на 30% лучше.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Khabarlak K., Koriashkina L. Fast Facial Landmark Detection and Applications: A Survey [Electronic resource]. – Journal of Computer Science and Technology, Volume 22, 2022. – Mode of access: https://journal.info.unlp.edu.ar/JCST/article/view/1972/1568. – Date of access: 16.05.2023.

2. Facial feature point detection: A comprehensive survey [Electronic resource] / Wang, Nannan, Xinbo Gao, Dacheng Tao, Heng Yang, Xuelong Li. – Neurocomputing 275, 2018. – pp. 50-65. – Mode of access: https://arxiv.org/pdf/1410.1037.pdf. – Date of access: 16.05.2023.

3. V. Kazemi and J. Sullivan. One millisecond face alignment with an ensemble of regression trees [Electronic resource]. – 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014. – pp. 1867-1874. – Mode of access: https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kazemi_One_Millisecond_Face_2014_CVPR_paper.pdf. – Date of access: 16.05.2023.

4. A New Dataset and Boundary-Attention Semantic Segmentation for Face Parsing/ Yinglu Liu, Hailin Shi, Hao Shen, Yue Si, Xiaobo Wang, Tao Mei. – The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-20), 2020. – 8p.

5. Deep Residual Learning for Image Recognition [Electronic resource] / Kaiming He [et al.]. – Mode of access: https://arxiv.org/pdf/1512.03385.pdf. – Date of access: 16.05.2023.

6. MobileNetV2: Inverted Residuals and Linear Bottlenecks [Electronic resource] / Mark Sandler [et al.]. – Mode of access: https://arxiv.org/pdf/1801.04381.pdf. – Date of access: 16.05.2023.

7. Squeeze-and-Excitation Networks [Electronic resource] / Jie Hu [et al.]. – Mode of access: https://arxiv.org/pdf/1709.01507.pdf. – Date of access: 16.05.2023.

8. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks [Electronic resource] / Mingxing Tan, Quoc V. Le. – Mode of access: https://arxiv.org/pdf/1905.11946.pdf. – Date of access: 16.05.2023.

9. Searching for MobileNetV3 [Electronic resource] / Andrew Howard [et al.]. – Mode of access: https://arxiv.org/pdf/1905.02244.pdf. – Date of access: 16.05.2023.

10. MobileViT: Light-weight, General-purpose, and Mobilefriendly Vision Transformer [Electronic resource] / Sachin Mehta, Mohammad Rastegari. – Mode of access: https://arxiv.org/pdf/2110.02178.pdf. – Date of access: 16.05.2023.

11. Wing Loss for Robust Facial Landmark Localisation with Convolutional Neural Networks [Electronic resource] / Zhen-Hua Feng [et al.]. – Mode of access: https://arxiv.org/pdf/1711.06753.pdf. – Date of access: 16.05.2023.