Отчет

По хакатону «Kryptonite ML Challenge»

Работу выполнила:
Команда MMG,
Студенты Финансового университета

Москва
2025

Поиск модели

Первым что мы сделали это отсмотрели большое количество SOTA-моделей распознавания лиц. Среди них:

- **FaceNet** (2015)
- ArcFace (2019)
- MagFace (2021)
- AdaFace (2022)
- GhostFace (2023)

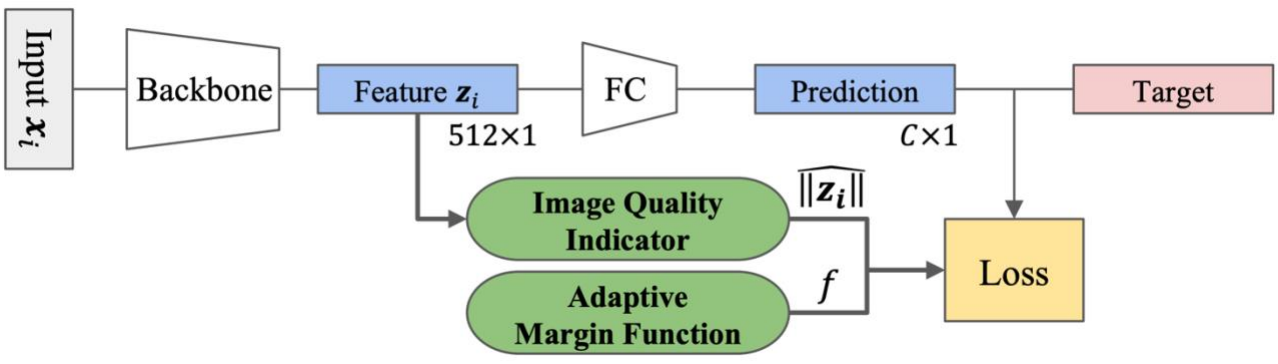
Сравнивали модели по популярным бенчмаркам и скорости работы. Среди бенчмарков были:

- LFW
- CPFLW
- CFPFP
- CALFW
- AGEDB
- IJBB
- IJBC
- TinyFace

*частично запускали сами, опирались на документации и статьи для сверки результатов и оценки решений.

Среди множества протестированных вариантов лучшими результатами по совокупности критериев (EER, скорость, устойчивость к «грязным» данным) выделилась модель AdaFace на базе ResNet101. Она динамически адаптирует маржинальную функцию, что особенно важно для «сложных» изображений.

Архитектура



Архитектура AdaFace

$$\mathcal{L}_{\text{AdaFace}}(x_i) = -\log \frac{\exp\left(s\left[\cos\left(\theta_{y_i} + g_{\text{angle}}(\|\hat{z}_i\|)\right) - g_{\text{add}}(\|\hat{z}_i\|)\right]\right)}{\exp\left(s\left[\cos\left(\theta_{y_i} + g_{\text{angle}}(\|\hat{z}_i\|)\right) - g_{\text{add}}(\|\hat{z}_i\|)\right]\right) + \sum_{j \neq y_i} \exp\left(s\cos\theta_j\right)}$$

Где:

$$g_{\text{angle}}(\|\hat{z}_i\|) = -m \cdot \|\hat{z}_i\|, \quad g_{\text{add}}(\|\hat{z}_i\|) = m \cdot \|\hat{z}_i\| + m, \quad \|\hat{z}_i\| = \text{clip}\left(\frac{\|z_i\| - \mu_z}{\sigma_z/h}, -1, 1\right).$$

AdaFace Loss

LFW	CPFLW	CFPFP	CALFW	AGEDB	IJBB@0.01	IJBC@0.01	TinyFace R1	TinyFace R5
99.82	95.65	99.30	95.93	98.10	96.55	97.82	76.10	78.92

Результаты работы AdaFace на популярных бенчмарках

Alignment

При работе с лицами важна правильная детекция лица и выравнивание (alignment). Даже самая точная модель может терять качество, если лицо «съехало» или неверно выделено. Поэтому мы перепробовали множество инструментов выравнивания.

Наиболее перспективными и запоминающимися были:

- **MTCNN** – быстрый, но плохая точность, много ошибок обнаружения лица.
- **RetinaFace** – отличное качество, но довольно медленный.
- **DFA** – Differentiable Face Aligner на базе ResNet50, по совокупным показателям (точность, скорость, удобство интеграции) оказался лучшим решением в нашем случае.

Работа с датасетами

Чтобы расширить датасет, взяли 10 000 персон, 200 000 фото из датасета CelebA и дополнительно сгенерировали дипфейки (по 4 на каждую персону) с помощью Roop, Ghost, Arc2Face, InstantID. Таким образом мы получили

максимально «разноплановый» набор для обучения и тестирования с разными архитектурами (диффузии, GAN, автоэнкодеры).

На каждого человека мы генерировали по 4 дипфейка, чтобы получить широкий спектр фальшивых изображений.

Формирование пар для обучения:

- Реальное лицо + то же лицо – метка 1 (50%)
- Реальное лицо + другое лицо – метка 0 (25%)
- Реальное лицо + его дипфейк – метка 0 (25%)

Такая пропорция помогла модели учиться различать разные лица, также как и выявлять синтетические подделки.

Алгоритм формирования пар

Мы берём комбинации реальных фото одного человека:

$$k = C_n^2$$

где n – количество реальных фото в папке.

Затем мы $k/2$ раз брали случайное фото из и этих n и случайное фото другого случайного человека – пары «реальное лицо + другое лицо».

После этого $k/2$ раз брали случайное лицо из тех же n и брали случайный дипфейк этого человека – пары «реальное лицо + его дипфейк».

Используя этот подход, мы обучили «претрейн» модель.

Ошибки в разметке

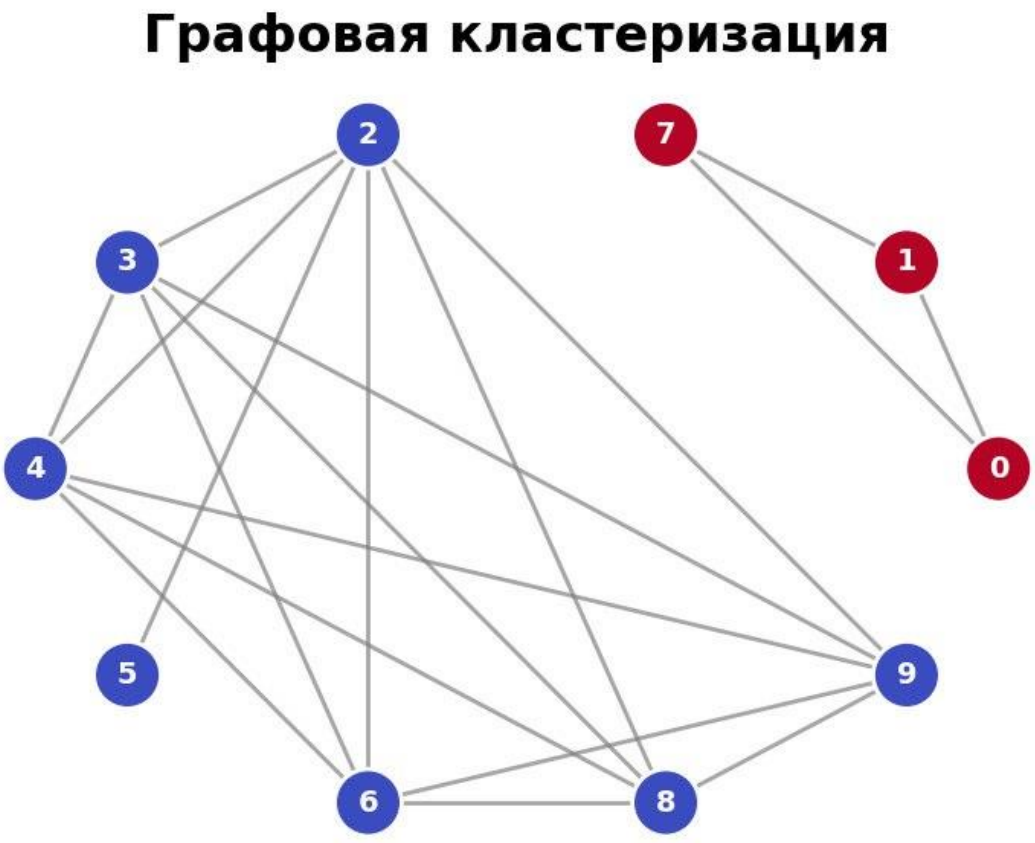
На выданном датасете же мы столкнулись с ошибочной разметкой в ~10% реальных фото. Проанализировав датасет, мы приняли решение провести его очистку.

Алгоритм следующий:

1. Используя исходную модель распознавания лиц AdaFace, для каждого набора реальных лиц в папке были построены матрицы сходства.

	0.jpg	1.jpg	2.jpg	3.jpg	4.jpg	5.jpg	6.jpg	7.jpg	8.jpg	9.jpg
0.jpg	1.000000	0.582046	-0.010843	-0.021038	-0.036042	-0.002022	0.027492	0.556213	-0.024669	-0.015600
1.jpg	0.582046	1.000000	-0.027603	-0.063650	-0.074505	-0.055325	-0.062826	0.916379	-0.077478	-0.078581
2.jpg	-0.010843	-0.027603	1.000000	0.770619	0.785005	0.579468	0.763535	-0.003984	0.772643	0.806771
3.jpg	-0.021038	-0.063650	0.770619	1.000000	0.852647	0.390069	0.793730	-0.070258	0.868653	0.855417
4.jpg	-0.036042	-0.074505	0.785005	0.852647	1.000000	0.414058	0.796396	-0.075372	0.851448	0.840460
5.jpg	-0.002022	-0.055325	0.579468	0.390069	0.414058	1.000000	0.457065	-0.013430	0.383513	0.395740
6.jpg	0.027492	-0.062826	0.763535	0.793730	0.796396	0.457065	1.000000	-0.048010	0.801057	0.799239
7.jpg	0.556213	0.916379	-0.003984	-0.070258	-0.075372	-0.013430	-0.048010	1.000000	-0.077718	-0.073854
8.jpg	-0.024669	-0.077478	0.772643	0.868653	0.851448	0.383513	0.801057	-0.077718	1.000000	0.859778
9.jpg	-0.015600	-0.078581	0.806771	0.855417	0.840460	0.395740	0.799239	-0.073854	0.859778	1.000000

2. На основе этих матриц формировался граф, в котором ребро между двумя лицами создавалось при высоком значении сходства и отсутствовало при низком. Порог специально брали больше, чем требовалось для того, чтобы не пропустить ошибки в разметке.



3. Далее была выполнена кластеризация графа – из него удалялись изолированные вершины, после чего он был разделён на отдельные компоненты связности.

4. После этого мы отсматривали глазами сложные и ошибочные случаи, которые выявила модель и корректировали их.

В данном случае всё верно, модель разделила двух разных людей в два кластера.

Таким образом, у нас получилось быстро и эффективно очистить датасет, отбросив очевидно верные случаи, что составляли большую часть датасета, около ~20%.

Обучение модели

В качестве loss для обучения мы пробовали множество различных вариантов. По кажому из использованных мы составили краткий список плюсов и минусов. Исходя из наших исследований, мы выяснили, что:

1) Adaptive Margin Loss:

+ адаптирует отступ (margin) динамически в зависимости от сходства между примерами, улучшая точность различения сложных случаев
+ увеличивает margin при низком сходстве (cos_sim), что позволяет модели эффективнее отталкивать негативные примеры и снижает влияние выбросов
+ уменьшает чувствительность к ошибочным меткам и аномальным примерам, обеспечивая стабильность и устойчивость обучения
– сложность подбора оптимальных параметров адаптации margin может привести к нестабильности обучения
– требует дополнительного вычислительного ресурса на расчет адаптивного отступа по сравнению с фиксированными margin

2) Contrastive Loss:

+ хорошо разделяет эмбеддинги настоящих и поддельных лиц
+ эффективно отталкивает дипфейки от оригиналов

– снижает компактность кластеров эмбеддингов одного и того же лица, что ухудшает качество распознавания

3) Circle Loss:

+ позволяет тонко контролировать сходство и различие между эмбеддингами
+ улучшает разделение дипфейков и оригиналов за счёт адаптивных границ принятия решений
– приводит к чрезмерному разнесению эмбеддингов, что негативно влияет на точность распознавания лиц

4) Center Loss:

– эмбеддингов одного класса вокруг общего центра, тем самым улучшая качество распознавание лиц
+ является менее эффективным в задаче отделения дипфейков, так как не создаёт чёткой границы между поддельными и оригинальными изображениями

5) Triplet Loss:

+ явно формулирует разницу расстояний между положительными и отрицательными примерами, помогая отделить дипфейки от реальных изображений
– генерация подходящих троек изображений является сложной задачей
– методика отрицательно влияет на стабильность и скорость сходимости обучения

6) Hard Negative Mining:

+ позволяет фокусироваться на сложных негативных примерах, усиливая различия между дипфейками и оригиналами в эмбеддинговом пространстве
– использование большого количества «жёстких» негативных примеров приводит к тому, что модель переобучается на отделении дипфейков, одновременно теряет обобщающую способность при распознавании лиц

7) Knowledge Distillation:

+ переносит знания распознавания лиц из предобученной модели-учителя – приближает эмбеддинги реальных лиц к эмбеддингам модели до обучения на дипфейках
– сильно ухудшает обучение детекции дипфейков

Резюмируя, многие loss-функции хорошо отталкивали дипфейки в другую часть эмбеддингового пространства, но при этом ухудшали качество распознавания лиц. Требовалось подобрать loss, который бы сохранял баланс между всеми факторами. В итоге мы остановились на адаптивном маржине (Adaptive margin).

На чистых данных – нашей валидационной выборке такой подход показал наилучшую сходимость. Буквально за 100 шагов с батчем 32 он уже достигал неплохих результатов, а через несколько сотен уже показывал стабильность на дипфейках. Мы приняли решение ориентироваться именно на него, ведь публичный лидерборд не отражал качество модели ввиду большого процента ошибочной разметки.

Тупиковые решения

В нашем исследовании мы старались пробовать, как и популярные, так и нестандартные подходы улучшения качества.

Предоставляем часть из них:

1) Аугументации, не меняющие признаки лица: световые, шум, повороты не дали ощутимого прироста по сравнению с базовым решением.

2) Автолейблинг публичного датасета – не дал прироста, вероятно из-за ощутимого количества ошибок в автоматической разметке.

3) Заморозка слоёв – всё кроме последних слоёв. Модель недостаточно эффективно адаптировалась к новым признакам дипфейков.

4) Добавляли в модель ветку с механизмом внимания (attention) – скорее всего, механизм внимания не давал преимуществ из-за избыточной сложности архитектуры.

5) Различные гиперпараметры, оптимизаторы, планировщики.

6) Обучение модель без претрейна на нашем датасете, только на выданном – модель плохо сходилась из-за ошибок в разметке.

Методика обучения

Мы обучали модель на нашем датасете, а после обученный претрейн использовали для подстройки под датасет хакатона.

В ходе обучения обнаружилось, что изображения недиффузионных моделей генерации дипфейков распознаются гораздо хуже, чем изображения диффузионных.

Качество распознавания дипфейков на InstanID и Arc2Face очень быстро достигало 99%+ точности, в то время как другие модели сходились гораздо медленнее.

Вероятно, это потому, что у диффузионных моделей очень узнаваемые паттерны из-за специфики итеративной генерации.

Лучшие гиперпараметры

cos θ — косинусное сходство

y ∈ {0, 1} — метка

s — масштабный коэффициент

$\mathbf{m_b}$ — базовый margin

α — кф. динамического масштабирования

1. Динамический **margin**:

$$m_d = m_b + \alpha \cdot (1 - \cos\theta)$$

2. Определение **логита**:

$$\text{logit} = s \cdot \left(\cos\theta - m_d \cdot y\right) = s \cdot \left(\cos\theta - \left(m_b + \alpha \cdot (1 - \cos\theta)\right) y\right)$$

3. Функция потерь **Adaptive Margin Loss** :

$$\mathcal{L} = -\left[y \cdot \log\left(\sigma(\text{logit})\right) + (1 - y) \cdot \log\left(1 - \sigma(\text{logit})\right)\right]$$

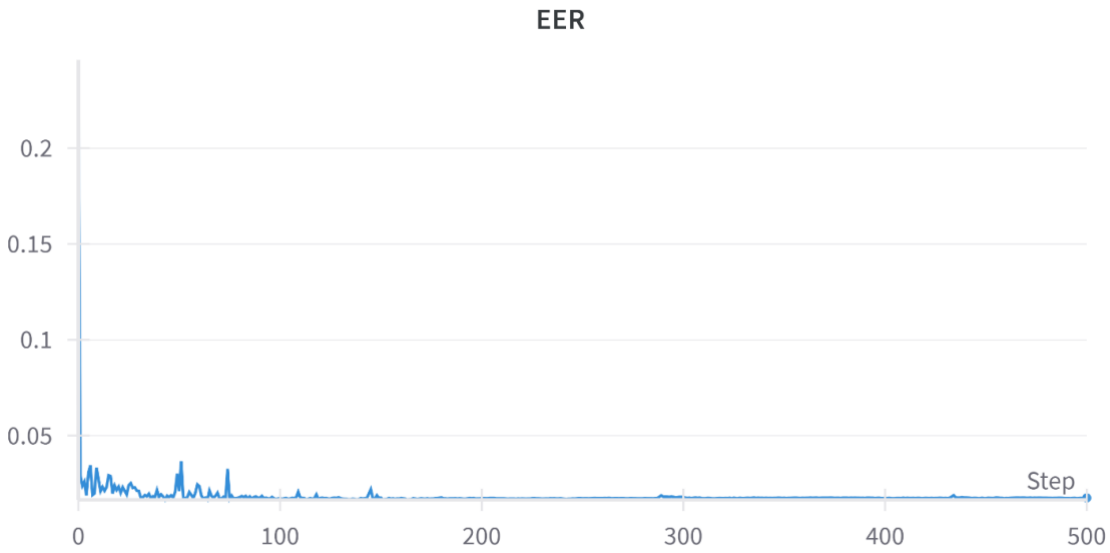
– Batch size: 32
– Learning rate: 5e-6
– Validation interval: 100
– Base_margin: 0.35
– Dynamic_scale: 0.1
– Weight_decay: 1e-2 – 1e-4

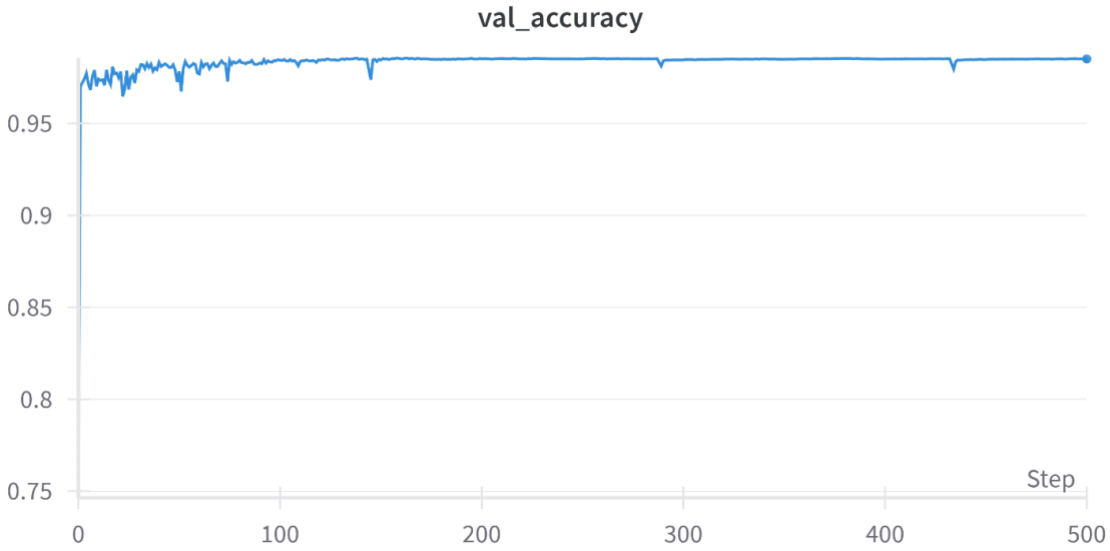
Оптимизатор AdamW:
– Learning Rate: 5e-6
– Weight decay: 1e-2

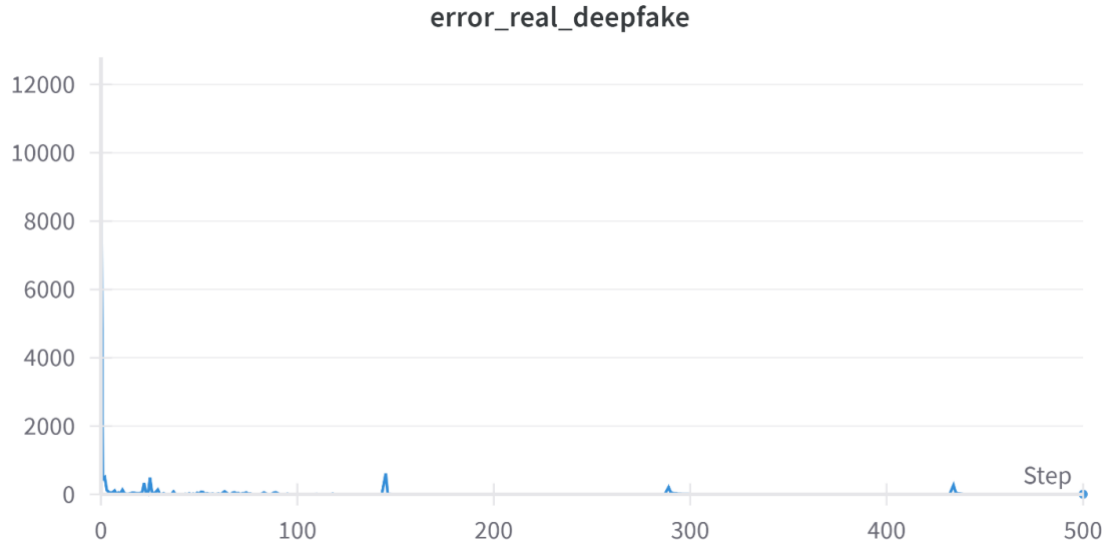
Scheduler ReduceLROnPlateau:
– Mode: min
– Factor: 0.5
– Patience: 10
– Threshold: 1e-4

Validation accuracy threshold: 0.2

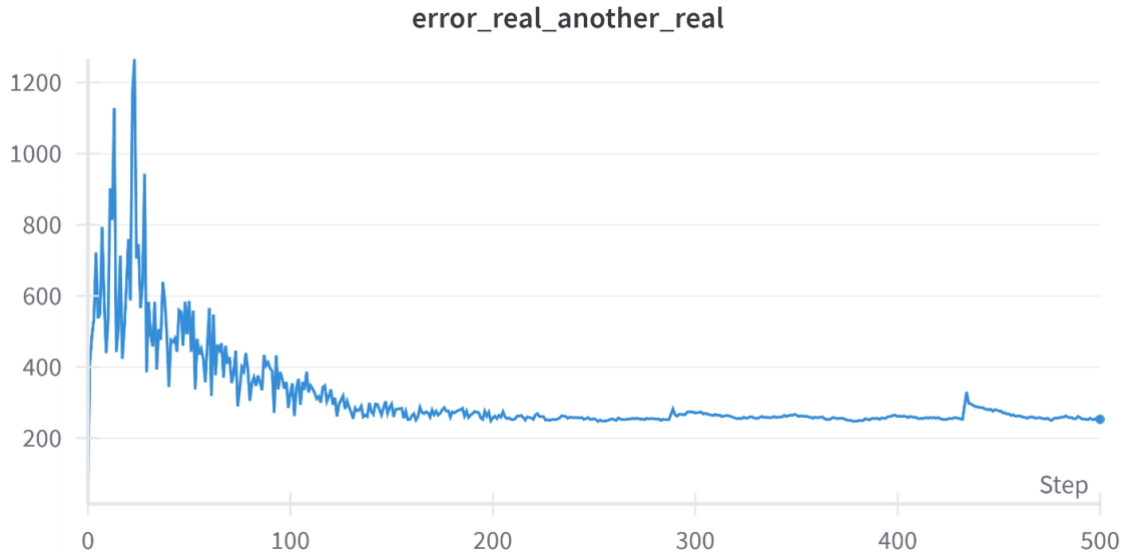
Графики обучения

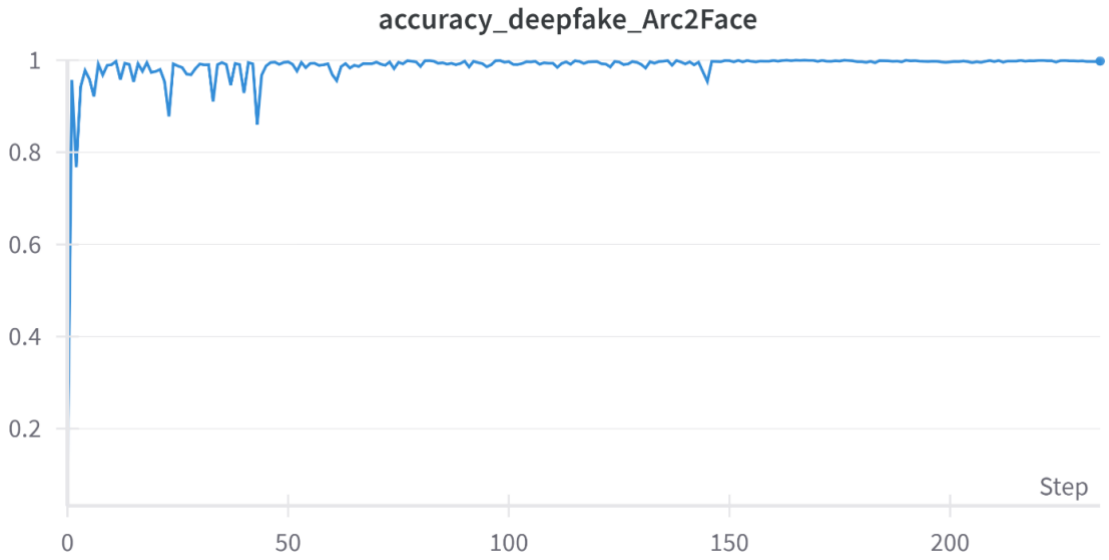


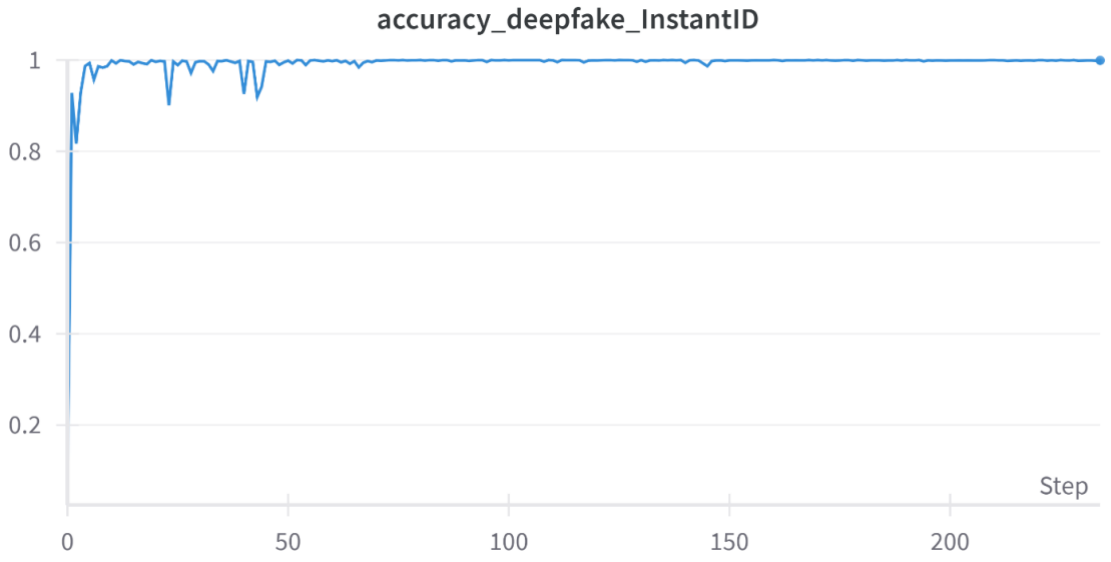


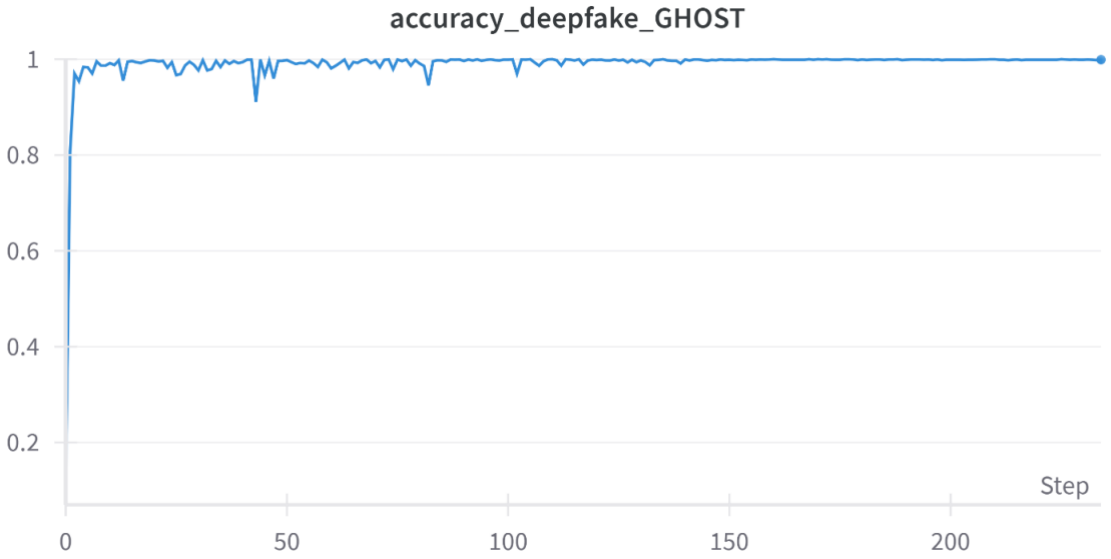


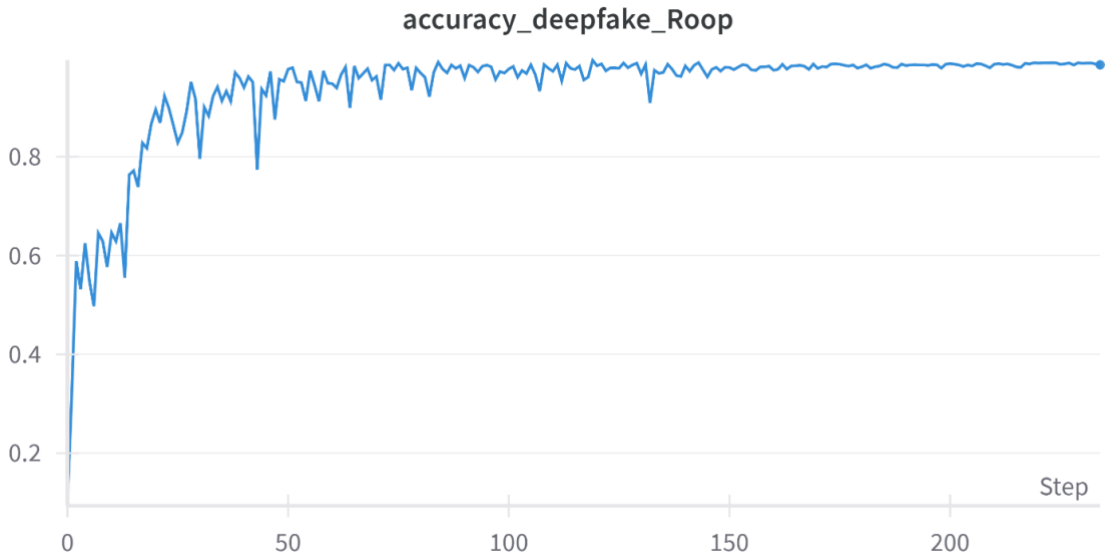

error_real_real
1200
1000
800
600
400
Step
0
100
200
300
400
500


error_real_another_real
1200
1000
800
600
400
200
Step
0
100
200
300
400
500


accuracy_deepfake_Arc2Face
1
0.8
0.6
0.4
0.2
Step
0
50
100
150
200


accuracy_deepfake_InstantID
1
0.8
0.6
0.4
0.2
Step
0
50
100
150
200


accuracy_deepfake_GHOST
1
0.8
0.6
0.4
0.2
Step
0
50
100
150
200


accuracy_deepfake_Roop
0.8
0.6
0.4
0.2
Step
0
50
100
150
200

Инференс

Процесс инференса был существенно ускорен благодаря оптимизации нескольких ключевых этапов обработки данных и расчетов. Мы применили параллельную загрузку и предварительную обработку изображений, что позволило снизить время простоя при чтении данных.

Для ускорения вычислений модель была переведена в формат ONNX, что обеспечило значительное повышение скорости работы за счёт эффективной реализации и оптимизации вычислений на GPU.

Дополнительно была реализована быстрая и точная процедура выравнивания лиц (alignment) с использованием модели DFA, работающей на GPU. Это позволило снизить задержки, связанные с препроцессингом и избежать узких мест при обработке большого количества данных.

Итоговый расчёт косинусного сходства эмбеддингов и сохранение результатов в CSV-файл также были оптимизированы. В итоге производительность решения существенно возросла, позволяя обрабатывать большие объёмы изображений за короткое время без потери точности и стабильности работы модели.

Результаты

Предложенное нами решение, основанное на модели AdaFace с архитектурой ResNet101 и адаптивной функцией потерь Adaptive Margin Loss, показало выдающиеся результаты на задаче распознавания и верификации лиц, в том числе и в условиях наличия дипфейков.

Проведённые эксперименты по оптимизации датасета, включая его очистку и расширение за счёт дополнительных реальных лиц, позволили значительно повысить стабильность и обобщающую способность модели. В частности, очистка ошибочной разметки позволила улучшить общие результаты.

Использование подхода с адаптивным margin обеспечило оптимальный баланс между способностью модели различать сложные реальные случаи и эффективно выявлять дипфейки. Этот подход показал наилучшую сходимость и стабильность на валидационной выборке, достигая высоких показателей за минимальное количество эпох.

Реализованные нами оптимизации препроцессинга и инференса позволили достичь скорости обработки изображений, подходящей для практического применения в реальных задачах.

Таким образом, модель AdaFace, дополненная эффективными подходами выравнивания и обработки данных, продемонстрировала высокую точность, устойчивость к сложным данным и эффективность на реальных и синтетических изображениях, подтверждая её перспективность для решения задач распознавания лиц и обнаружения дипфейков.