

**«Применение методов компьютерного зрения для
автоматизированного анализа медицинских изображений»**

Выполнил(а):

Вол М. Л.

Москва 2025

Аннотация.

В работе рассматривается задача автоматического выявления патологий на рентгенограммах опорно-двигательного аппарата на основе датасета MURA. Предложен пайплайн обучения нейросетевых классификаторов в постановке *per-area* (отдельно по анатомическим областям), включающий проверку структуры данных, предобработку и аугментацию. В качестве базовой архитектуры использована DenseNet, как эффективная сверточная сеть с плотными связями для устойчивого распространения признаков и градиентов. Для повышения обобщающей способности применены методы регуляризации на уровне данных MixUp и CutMix. Реализована интерпретируемость и анализ внимания модели с использованием подходов семейства Class Activation Mapping (на практике - Grad-CAM), а также исследована стратегия подавления наиболее дискриминативных областей (CAM-erasing) как способ смещения внимания модели к альтернативным информативным признакам. Выполнена гиперпараметрическая оптимизация с использованием Optuna и реализовано двухфазное дообучение финальной модели. Дополнительно рассмотрен гибридный подход с трансформерным компонентом на базе Swin Transformer для усиления представлений при сложных областях. Качество моделей оценивается по ROC AUC и дополнительным метрикам классификации на *image-level* и *study-level*, проводится сравнительный анализ подходов и разбор типичных ошибок по областям. Для финальной DenseNet получено значение ROC AUC = 0.9038 (macro, *study-level*) и ROC AUC = 0.8855 (macro, *image-level*); по анатомическим областям *study-level* ROC AUC находится в диапазоне 0.8624 - 0.9333 (лучшая область - XR_HUMERUS, наиболее сложная - XR_HAND).

Ключевые слова.

MURA; рентгенография; опорно-двигательный аппарат; musculoskeletal radiographs; обнаружение патологий; выявление аномалий; медицинские изображения; глубокое обучение; сверточные нейронные сети (CNN); DenseNet; transfer learning; fine-tuning; аугментация данных; регуляризация; MixUp; CutMix; class activation mapping (CAM); Grad-CAM; интерпретируемость;

гиперпараметрическая оптимизация; Optuna; двухфазное обучение; per-area обучение; image-level; study-level; ROC AUC; precision-recall; confusion matrix; калибровка вероятностей; надежностные диаграммы; Expected Calibration Error (ECE); Swin Transformer; гибридные модели.

Объект и предмет исследования.

Объект исследования - рентгенографические изображения опорно-двигательного аппарата (датасет MURA) и процесс автоматического выявления на них патологических изменений.

Предмет исследования - методы и алгоритмы глубокого обучения для бинарной классификации рентгенограмм по признаку «норма/патология», включая архитектуры DenseNet и гибридные решения с трансформерами, а также методы улучшения качества и интерпретируемости (аугментации, регуляризация, гиперпараметрическая оптимизация, CAM/Grad-CAM).

Оглавление

1. Введение.....	5
2. Задача исследования.....	6
3. Цель исследования.....	6
4. Набор данных и предобработка.....	6
4.1 Описание датасета MURA	6
4.2 Проверка структуры и загрузка данных.....	8
4.3 Предобработка данных	10
4.4 Аугментация данных.....	11
5. Архитектура и инфраструктура моделей	13
5.1 Общая архитектура DenseNet	14
5.2 Пер-областные загрузчики данных	15
5.3 Инфраструктура обучения	16
5.4 САМ-инфраструктура	18
6. Методы и подходы к обучению моделей.....	24
6.1 Подход 1 - Baseline DenseNet.....	24
6.2 Подход 2 - DenseNet с улучшениями	27
6.2.1 Продвинутые аугментации	28
6.2.2 Модифицированный цикл обучения.....	30
6.2.3 Гиперпараметрическая оптимизация.....	32
6.3 Подход 3 - Финальная DenseNet (двухфазное обучение).....	35
6.4 Подход 4 - Гибридная модель DenseNet + Swin Transformer.....	39
7. Экспериментальные результаты	43
7.1 Метрики оценки качества	44
7.2 Результаты baseline-модели	46
7.3 Результаты модели с НРО и САМ	62
7.4 Результаты финальной DenseNet	80
7.5 Результаты гибридной модели	97
8. Сравнительный анализ моделей.....	108
9. Выводы.....	110
10. Заключение и направления дальнейших исследований	111

Введение

Разработка методов автоматизированного анализа рентгенограмм опорно-двигательного аппарата представляет практический интерес для задач первичного скрининга и поддержки принятия врачебных решений. В постановке «норма/патология» корректная интерпретация рентгенологического исследования позволяет исключить заболевание и, тем самым, снизить потребность в дополнительных диагностических процедурах и вмешательствах.

Прогресс глубокого обучения в компьютерном зрении в значительной степени связан с появлением больших размеченных наборов данных, используемых как для обучения, так и для сопоставимого сравнения методов. В медицинской визуализации аналогичная потребность в крупных и качественно размеченных данных особенно выражена из-за сложности разметки и высокой стоимости экспертного труда.

Одним из наиболее известных открытых наборов данных для исследования данной задачи является MURA (Musculoskeletal Radiographs), представленный группой Stanford: датасет включает 14863 исследования верхней конечности, каждое из которых содержит один или несколько снимков и имеет экспертную разметку «норма/патология». Анализ исходной работы подчеркивает клиническую значимость задачи, связывая ее с высокой распространенностью заболеваний опорно-двигательного аппарата и их вкладом в боль и инвалидизацию. Авторы также фиксируют, что даже сильные базовые модели демонстрируют неоднородное качество по анатомическим областям, что делает задачу релевантной для дальнейших исследований.

В настоящей работе исследуется построение и улучшение моделей выявления патологий на рентгенограммах MURA в постановке обучения по анатомическим областям. Рассматривается базовая архитектура DenseNet и методы повышения обобщающей способности, а также методы интерпретации решений на основе карт активаций и стратегия перераспределения внимания. Качество моделей оценивается по ROC AUC и дополнительным метрикам бинарной классификации на уровнях

image-level и study-level, что позволяет провести сопоставимый анализ по областям и разбор типичных ошибок.

Задача исследования

Разработать и экспериментально проверить улучшенный по сравнению с базовым решением авторов MURA подход к автоматическому выявлению патологий на рентгенограммах опорно-двигательного аппарата в постановке бинарной классификации «норма/патология» с обучением по анатомическим областям (per-area), включающий предобработку данных, построение и обучение моделей, интерпретацию результатов и сравнительный анализ качества.

Цель исследования

Достичь более высокого качества выявления патологий на датасете MURA по метрике ROC AUC (на уровнях study-level и image-level) по сравнению с опубликованным базовым результатом, представленным командой Stanford, за счет применения усовершенствованного пайплайна обучения (аугментации и регуляризация, гиперпараметрическая оптимизация, двухфазное дообучение, методы интерпретируемости CAM/Grad-CAM и гибридные архитектурные решения).

Набор данных и предобработка

Описание датасета MURA

В работе используется датасет MURA (Musculoskeletal Radiographs), предложенный исследовательской группой Stanford для задачи выявления патологий на рентгенограммах верхней конечности. Датасет представляет данные в формате исследований: каждое исследование включает один или несколько снимков и имеет бинарную метку normal/abnormal, присвоенную на уровне исследования (см. Рисунок 1).

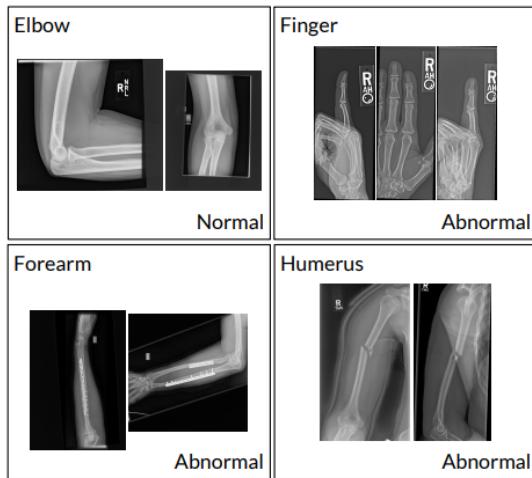


Рисунок 1 - Примеры снимков датасета

Согласно исходной публикации, MURA содержит 14863 исследования от 12173 пациентов и 40561 мультиакурсное изображение. Данные собраны из клинической системы хранения изображений PACS¹ Stanford Hospital в рамках исследования, одобренного IRB²; изображения де-идентифицированы и соответствуют требованиям НИРАА³. Разметка выполнена сертифицированными радиологами в процессе клинической интерпретации (период 2001-2012).

Все исследования относятся к одной из семи стандартных анатомических областей верхней конечности: elbow, finger, forearm, hand, humerus, shoulder, wrist, что позволяет анализировать данные как в общей постановке, так и по областям. В статье также приведена сводная статистика классов: 9045 исследований отнесены к normal и 5818 - к abnormal.

¹ PACS (Picture Archiving and Communication System) - система архивации и передачи медицинских изображений, предназначенная для хранения, поиска и обмена исследованиями (рентген, КТ, МРТ и др.) в медицинских организациях.

² RB (Institutional Review Board) - независимый этический комитет, который рассматривает и одобряет исследования с участием людей и/или использованием их медицинских данных, оценивая этические риски и соблюдение прав участников.

³ HIPAA (Health Insurance Portability and Accountability Act) - федеральный закон США, устанавливающий требования к защите конфиденциальности и безопасности медицинской информации; в контексте исследований обычно подразумевает де-идентификацию данных и соблюдение правил обращения с защищенной медицинской информацией (PHI).

Study	Train		Validation		Total
	Normal	Abnormal	Normal	Abnormal	
Elbow	1094	660	92	66	1912
Finger	1280	655	92	83	2110
Hand	1497	521	101	66	2185
Humerus	321	271	68	67	727
Forearm	590	287	69	64	1010
Shoulder	1364	1457	99	95	3015
Wrist	2134	1326	140	97	3697
Total No. of Studies	8280	5177	661	538	14656

Рисунок 2 - Состав датасета MURA по классам (normal/abnormal) и анатомическим областям верхней конечности (shoulder, humerus, elbow, forearm, wrist, hand, finger)

Авторы исходной работы используют разбиение без пересечения пациентов между подмножествами: training - 11184 пациента / 13457 исследований / 36808 изображений, validation - 783 пациента / 1199 исследований / 3197 изображений, test - 206 пациентов / 207 исследований / 556 изображений (см. Рисунок 2). Для тестового набора дополнительно собраны повторные разметки от шести радиологов Stanford; «золотой стандарт» формировался как majority vote выбранной тройки экспертов, что использовалось для сравнения моделей и оценки согласия.

Для понимания клинической неоднородности положительного класса в статье выполнен разбор 100 abnormal исследований по текстам заключений: среди наиболее частых находок упоминаются переломы, металлоконструкции, дегенеративные изменения и другие аномалии (например, поражения/подвывихи).

Проверка структуры и загрузка данных

На первом этапе выполнена проверка корректности структуры датасета в рабочем каталоге ROOT. В каталоге присутствуют поддиректории train/ и valid/, а также вспомогательные CSV-файлы со списками путей к изображениям (train_image_paths.csv, valid_image_paths.csv) и разметкой на уровне исследований (train_labeled_studies.csv, valid_labeled_studies.csv). Наличие всех ключевых элементов подтверждено программной проверкой существования файлов и каталогов.

Для загрузки данных реализован класс MURADataset, который считывает пути к изображениям из CSV-файла (по одной записи на изображение) и формирует выборку для обучения и валидации. В процессе инициализации выполняется

нормализация путей: приведение разделителей к единому виду, корректная обработка относительных и абсолютных путей и сборка полного пути относительно ROOT (см. Рисунок 3). Это обеспечивает воспроизводимую работу кода независимо от исходного формата путей в CSV. При отсутствии путей возбуждается исключение, что позволяет выявлять ошибки структуры данных на ранней стадии.

```
for p in paths_raw:
    p = p.replace('\\', '/')
    if p.startswith('train/') or p.startswith('valid/'):
        full = os.path.join(root_dir, *p.split('/'))
    else:
        if '/train/' in p:

            tail = p.split('/train/', 1)[1]
            full = os.path.join(root_dir, 'train', *tail.split('/'))
        elif '/valid/' in p:
            tail = p.split('/valid/', 1)[1]
            full = os.path.join(root_dir, 'valid', *tail.split('/'))
        else:

            full = os.path.join(root_dir, p)
    normed.append(os.path.normpath(full))
```

Рисунок 3 - нормализация путей в классе датасета

Метка класса формируется автоматически на основе структуры MURA: если путь к изображению содержит подстроку positive⁴, объект относится к классу патологии (1), иначе - к норме (0). Это обеспечивает согласованность image-level разметки с исходной файловой организацией датасета.

Построены загрузчики данных для обучающей и валидационной выборок с унифицированными преобразованиями: приведение изображений к фиксированному размеру 224×224⁵ и преобразование в тензор (см. Рисунок 4). Для контроля корректности загрузки дополнительно проверялась форма батча.

Batch shape: (16, 3, 224, 224) labels sample: [1, 1, 0, 1, 1, 0, 1, 1, 0, 0]
--

⁴ Вложенность MURA, пример пути: /train/XR_SHOULDER/patient00001/study1_positive/image1.png'

⁵ Размер 224×224 выбран в соответствии с рекомендованными преобразованиями для предобученных весов DenseNet121 в библиотеке Torchvision: изображения приводятся к resize_size=256 с последующим центральным кадрированием до crop_size=224, после чего выполняется нормализация.

```

train_loader, valid_loader, train_ds, valid_ds = make_loaders(ROOT)

print("Train images:", len(train_ds), " | class balance:", Counter(train_ds.labels))
print("Valid images:", len(valid_ds), " | class balance:", Counter(valid_ds.labels))

Train images: 36808 | class balance: Counter({0: 21935, 1: 14873})
Valid images: 3197 | class balance: Counter({0: 1667, 1: 1530})

```

Рисунок 4 - Размеры выборок и баланс классов

Для последующего обучения в постановке per-area реализовано извлечение анатомической области из пути к изображению. Подсчет количества изображений по областям показал ожидаемое распределение: доминируют области XR_WRIST и XR_SHOULDER, а наименьшее число изображений представлено в XR_HUMERUS и XR_FOREARM.

```

Train per area: Counter({'XR_WRIST': 9752, 'XR_SHOULDER': 8379, 'XR_HAND': 5543,
'XR_FINGER': 5106, 'XR_ELBOW': 4931, 'XR_FOREARM': 1825, 'XR_HUMERUS': 1272})

Valid per area: Counter({'XR_WRIST': 659, 'XR_SHOULDER': 563, 'XR_ELBOW': 465,
'XR_FINGER': 461, 'XR_HAND': 460, 'XR_FOREARM': 301, 'XR_HUMERUS': 288})

```

Предобработка данных

Предобработка данных направлена на приведение изображений MURA к единому формату, обеспечивающему корректную подачу в нейросетевую модель и воспроизводимость экспериментов. Для обеспечения сопоставимости результатов и построения моделей в постановке per-area изображения фильтруются и загружаются отдельно для каждой области. Входной размер изображений фиксируется как IMG_SIZE = 224 и упрощает применение transfer learning. На этапе базовой предобработки выполняются преобразования: конвертация изображения в формат RGB и приведение к заданному размеру, после чего изображение переводится в тензор.

Для загрузки данных из исходной иерархии MURA реализован датасет MURAFolderDataset, который осуществляет обход файловой структуры по шаблону

```
область → пациент → исследование → изображения
```

Дополнительно реализована фильтрация файлов по допустимым расширениям изображений для исключения технических и скрытых файлов, не относящихся к данным.

```
IMG_EXTS = {'.png', '.jpg', '.jpeg', '.bmp', '.tif', '.tiff'}
```

Для повышения надежности пайплайна выполняется проверка читаемости файлов изображений (`PIL.Image.verify()`): поврежденные или некорректные файлы пропускаются, что предотвращает ошибки на этапе обучения и валидации (см. Рисунок 5). При отсутствии изображений в выбранной области возбуждается исключение, позволяющее диагностировать проблемы структуры данных на раннем этапе.

```
for imgp in sorted(study_dir.iterdir()):
    if imgp.is_file() and imgp.suffix.lower() in IMG_EXTS and not imgp.name.startswith("."):
        try:
            with Image.open(imgp) as im:
                im.verify()
            paths.append(str(imgp))
            labels.append(label)
        except Exception:
            continue
```

Рисунок 5 – Обработка читаемости изображения

Такая схема предобработки обеспечивает

- (1) унификацию входных данных для всех областей,
- (2) корректное извлечение разметки из структуры MURA,
- (3) устойчивость к поврежденным файлам,
- (4) возможность воспроизводимости экспериментов для каждой анатомической области.

Аугментация данных

```
transform_train = transforms.Compose([
    transforms.RandomResizedCrop(224, scale=(0.9, 1.1)),
    transforms.RandomRotation(degrees=15),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.ColorJitter(brightness=0.1, contrast=0.1),
```

```
transforms.ToTensor(),  
transforms.Normalize(mean=[0.485,0.456,0.406],  
std=[0.229,0.224,0.225]),  
])  
transform_valid = transforms.Compose([  
    transforms.Resize((224,224)),  
    transforms.ToTensor(),  
    transforms.Normalize(mean=[0.485,0.456,0.406],  
std=[0.229,0.224,0.225]),  
])
```

Для повышения устойчивости модели к вариативности изображений и снижения риска переобучения в обучающей выборке применяется аугментация данных - стохастические преобразования входных изображений, моделирующие допустимые изменения условий съемки и положения объекта. Аугментации применяются только на этапе обучения, тогда как для валидации используются детерминированные преобразования, обеспечивающие корректное и воспроизводимое сравнение моделей.

Для обучающей выборки сформирован конвейер преобразований `transform_train`, включающий следующие операции:

- Случайное кадрирование с приведением к размеру 224×224 , что имитирует небольшие изменения масштаба и положения объекта в кадре;
- Случайный поворот изображения в допустимом диапазоне, повышающий инвариантность к наклону при укладке конечности;
- Случайное отражение по горизонтали, используемое как простая форма симметрийной регуляризации;
- Небольшие изменения яркости и контраста, моделирующие вариативность экспозиции и параметров обработки снимка.

После геометрических и фотометрических преобразований выполняется преобразование в тензор и нормализация по статистикам ImageNet, что соответствует стандартной процедуре использования предобученных весов сверточных сетей в постановке transfer learning.

Для валидационной выборки используется `transform_valid`, включающий только приведение к фиксированному размеру 224×224 , преобразование в тензор и нормализацию теми же статистиками ImageNet. Отсутствие случайных преобразований на валидации помимо обеспечивания воспроизводимости метрик, открывает доступ к возможности корректности сравнения различных конфигураций обучения.

Описанные аугментации позволяют увеличить эффективное разнообразие обучающих данных, улучшить обобщающую способность модели и повысить устойчивость к типичным вариациям рентгенографических изображений, не изменяя при этом семантическую принадлежность к классу.

Архитектура и инфраструктура моделей

В данной главе описываются выбранная архитектура нейросетевой модели и элементы экспериментальной инфраструктуры, обеспечивающие воспроизводимость обучения и корректный сравнительный анализ. В рамках работы используются модели, обучаемые отдельно для каждой анатомической области, поэтому особое значение имеет единообразная процедура построения архитектуры, загрузки предобученных весов, настройки выхода под бинарную классификацию, а также механизмы обучения, сохранения чекпоинтов и последующего анализа результатов. Отдельное внимание уделяется поддержке интерпретируемости решений с использованием карт активаций (CAM/Grad-CAM), так как медицинские приложения требуют помимо высокой точности возможность анализа причин принятого решения.

Общая архитектура DenseNet

В качестве базовой архитектуры для решения задачи бинарной классификации «норма/патология» выбрана DenseNet-121 (см. Рисунок 6). DenseNet относится к классу сверточных нейросетей с плотными связями: выход каждого слоя передается на вход всех последующих слоев в пределах блока. Такая схема способствует повторному использованию признаков, более стабильному распространению градиентов и часто позволяет получать высокое качество при сравнительно умеренном числе параметров. Эти свойства были сформулированы в оригинальной работе Huang и соавт., где DenseNet показала эффективность на задачах распознавания изображений.

```
def build_model():
    m = models.densenet121(weights=models.DenseNet121_Weights.IMGNET1K_V1)
    in_feats = m.classifier.in_features
    m.classifier = nn.Linear(in_feats, 1)
    return m.to(DEVICE)
```

Рисунок 6 - Выбранная архитектура нейросетевой модели

Выбор обусловлен еще одной причиной. Использование DenseNet обеспечивает сопоставимость с базовым ориентиром в исходной статье по MURA: авторы датасета Stanford обучали 169-слойную DenseNet как baseline-модель для детекции и локализации аномалий на MURA. В настоящей работе применяется DenseNet-121, что сохраняет архитектурную идеологию baseline Stanford, но снижает вычислительную сложность: модель имеет меньше параметров и требует меньше памяти, что особенно важно при проведении серии экспериментов по нескольким анатомическим областям (per-area), гиперпараметрической оптимизации и двухфазного дообучения. Такой выбор представляет собой компромисс между качеством и ресурсами, позволяя увеличить число воспроизводимых прогонов и ускорить цикл экспериментов без перехода к принципиально иной архитектуре.

Модель инициализируется предобученными весами ImageNet (IMAGENET1K_V1), что позволяет использовать transfer learning и ускоряет

сходимость. Для адаптации под бинарную постановку стандартный классификатор DenseNet заменяется на линейный слой, выдающий один логит:

- DenseNet121 (pretrained) → извлекает признаки изображения;
- Linear(in_features → 1) → формирует скалярный выход (логит) для класса «патология»

При обучении выходной логит интерпретируется как вероятность класса, а оптимизация выполняется с использованием стандартной функции потерь для бинарной классификации, объединяющей логит и ф. активации в численно устойчивой форме.

Пер-областные загрузчики данных

Датасет MURA включает исследования разных анатомических областей, для которых характерны различающиеся визуальные паттерны, распределения классов и сложность выявления патологий. В связи с этим в работе используется постановка per-area: для каждой области (XR_WRIST, XR_SHOULDER и т.д.) формируется отдельная обучающая и валидационная выборка и далее обучается отдельная модель/конфигурация.

Для унификации экспериментов реализована функция `make_loaders_for_area`(см. Рисунок 7), которая:

- строит пути к train/ и valid/ внутри ROOT;
- создает датасеты MURAFolderDataset (см. главу 4.3) только для выбранной области area;
- применяет разные преобразования для обучения и валидации (`transform_train`, `transform_valid`) (см. главу 4.4);
- формирует DataLoader для train (`shuffle=True`) и valid (`shuffle=False`).

```

def make_loaders_for_area(area, return_path_valid=False):
    train_dir = os.path.join(ROOT, 'train')
    valid_dir = os.path.join(ROOT, 'valid')

    ds_tr = MURAFolderDataset(train_dir, area=area, transform=transform_train, return_path=False)
    ds_va = MURAFolderDataset(valid_dir, area=area, transform=transform_valid, return_path=return_path_valid)

    dl_tr = DataLoader(ds_tr, batch_size=BATCH_SIZE, shuffle=True, num_workers=NUM_WORKERS, pin_memory=True)
    dl_va = DataLoader(ds_va, batch_size=BATCH_SIZE, shuffle=False, num_workers=NUM_WORKERS, pin_memory=True)
    return ds_tr, ds_va, dl_tr, dl_va

```

Рисунок 7 – сбор загрузчиков по областям

Параметр `return_path_valid` используется для задач, где помимо изображения и метки требуется путь к файлу. Это важно для вычисления study-level метрик (через идентификатор в пути) и построения CAM/Grad-CAM (привязка карт внимания к исходным изображениям).

Инфраструктура обучения

Взвешивание классов (борьба с дисбалансом)

В данных MURA доли классов `normal`/`abnormal` могут отличаться как в целом, так и по отдельным областям. Чтобы редкий класс не «терялся» при оптимизации, реализована вспомогательная функция `class_weight_from_counts`, вычисляющая веса классов обратно пропорционально их частоте (см. Рисунок 8).

```

def class_weight_from_counts(labels):
    c = Counter(labels)
    total = sum(c.values())
    w0 = total / (2.0 * max(1, c.get(0,0)))
    w1 = total / (2.0 * max(1, c.get(1,0)))
    return torch.tensor([w0, w1], dtype=torch.float32).to(DEVICE)

```

Рисунок 8 - Функция взвешивания классов

Эти веса далее будут использоваться в функции потерь (и в схеме с `pos_weight` для ВСЕ), обеспечивая более сбалансированный вклад классов в градиенты.

Извлечение идентификатора исследования

Поскольку в MURA разметка задана на уровне исследования (study), важно корректно объединять несколько снимков одного исследования. Для этого реализована функция `study_id_from_path` (см. Рисунок 9), которая извлекает идентификатор вида:

patientXXXX/studyY_positive|negative

из полного пути:

.../XR_AREA/patientXXXX/studyY_*/image.png.

В результате, все кадры одного исследования будут агрегироваться под одним study_id.

```
def study_id_from_path(p: str) -> str:
    parts = Path(p).as_posix().split('/')

    idx = None
    for i, s in enumerate(parts):
        if re.fullmatch(r'XR_[A-Z]+', s or ''):
            idx = i
            break

    if idx is None or idx + 2 >= len(parts):
        return "UNK/UNK"

    patient = parts[idx+1]
    study   = parts[idx+2]
    return f"{patient}/{study}"
```

Рисунок 9 – Функция извлечения идентификатора исследования

Оценка на image-level и study-level

Для корректного сравнения моделей реализована функция evaluate_study(model, loader, desc="valid_study"), которая вычисляет метрики:

на уровне изображений (image-level): лосс, accuracy, ROC AUC;

на уровне исследований (study-level): accuracy, ROC AUC, F1.

На этапе валидации сохраняются вероятности $p = \text{sigmoid}(\text{logit})$ для каждого изображения и соответствующие пути. Далее данные агрегируются по study_id:

- вероятности усредняются внутри исследования;
- истинная метка исследования восстанавливается из image-level меток (через усреднение и округление до 0/1);
- после этого вычисляются метрики на уровне исследований.

Все это обусловлено тем, что решение должно быть устойчивым не для отдельного кадра, а для исследования целиком (где кадры - разные ракурсы одной и той же клинической ситуации).

САМ-инфраструктура

Для повышения интерпретируемости моделей и анализа того, какие области изображения влияют на предсказание, реализована инфраструктура построения карт активаций класса (Class Activation Mapping, CAM). В работе используется вариант семейства CAM - Grad-CAM++, позволяющий получать карту важности даже в случаях, когда классическая Grad-CAM дает размытые или слабые отклики.

Построение карты внимания для одного изображения

Базовая функция `get_cam_auto(model, x, layer_paths=("features.denseblock4", "features.denseblock3"), tta_hflip=True, relu=False, border_crop=0.0)` вычисляет CAM для одного входного изображения x размера $(1, C, H, W)$ и возвращает:

cam - карту важности, нормированную в диапазон $[0, 1]$;

$prob$ - вероятность класса $abnormal$ (через $\text{sigmoid}(\text{logit})$).

Для корректной интерпретации CAM учитывается знак целевого класса: при $prob \geq 0.5$ карта строится в направлении, усиливающем положительный класс (*positive*), иначе - в направлении отрицательного.

```
with torch.no_grad():
    prob = torch.sigmoid(model(x).squeeze(0)).item()
    score_sign = +1.0 if prob >= 0.5 else -1.0
```

Это реализовано через множитель $score_sign \in \{+1, -1\}$, определяющий, какой логит используется для обратного распространения градиента.

Выбор слоя и fallback-стратегия

CAM вычисляется относительно сверточного слоя DenseNet. В качестве приоритетного источника используется последний плотный блок features.denseblock4.

```
def _cam_for_layer(layer_path, x_in):
    cammer = _GradCamPP(model, _resolve_module(model,
layer_path))

    cam = cammer.cam(x_in, score_sign=score_sign,
relu=relu)
    cammer.remove()
    return cam

for lp in layer_paths:
    cam = _cam_for_layer(lp, x)
    if (cam >= 0.2).sum() > 0:
        break
```

Если полученная карта оказывается практически пустой (эвристика: отсутствуют пиксели с интенсивностью ≥ 0.2), выполняется fallback на более ранний блок features.denseblock3. Такая стратегия снижает вероятность получения неинформативных карт на отдельных изображениях и повышает стабильность визуализации.

Устойчивость CAM за счет TTA

Для повышения устойчивости карт внимания применяется test-time augmentation в виде горизонтального отражения.

```
if tta_hflip:
    x_fl = torch.flip(x, dims=[3])
    cam_fl = None
    for lp in layer_paths:
        cam_fl = _cam_for_layer(lp, x_fl)
        if (cam_fl >= 0.2).sum() > 0:
            break
    cam = 0.5 * (cam + np.fliplr(cam_fl))
```

CAM строится для исходного изображения и для его hflip-версии, после чего карта для отраженного изображения разворачивается обратно и усредняется с исходной. Это уменьшает чувствительность CAM к случайным особенностям входа и делает визуализации более стабильными.

Реализация Grad-CAM++ через хуки

Вычисление Grad-CAM++ реализовано классом `_GradCamPP`, который регистрирует:

`forward_hook` для сохранения активаций выбранного слоя (`acts`);

`backward_hook` для сохранения градиентов по этим активациям (`grads`).

```
class _GradCamPP:
    def __init__(self, model, target_module):
        self.model = model
        self.t = target_module
        self.acts = None
        self.grads = None
        self.ha = self.t.register_forward_hook(self._fh)
        self.hg = self.t.register_full_backward_hook(self._bh)
```

Далее CAM++ формируется как взвешенная сумма карт признаков слоя, где веса каналов вычисляются по формуле Grad-CAM++⁶ на основе градиентов (учитываются положительные градиенты и α -коэффициенты), после чего карта нормируется в $[0,1]$.

- A^k - k -я карта признаков (активация) выбранного слоя, A_{ij}^k - значение в позиции $(i; j)$
- y^c - «score» (логит) класса c
- $\frac{\partial y^c}{\partial A_{ij}^k}$ - градиент по активациям

1) α - коэффициенты (Grad-CAM++):

⁶ Речь про стандартную формулу Grad-CAM++ (Chattpadhy et al., 2018). Она отличается от обычного Grad-CAM тем, что веса каналов считаются не просто средним градиентом, а через α -коэффициенты, зависящие от 2-й и 3-й производных (в реализации — квадраты/кубы градиентов).

$$\alpha_{ij}^{kc} = \frac{\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)^2}{2\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)^2 + \Sigma_a \Sigma_b A_{ab}^k \left(\frac{\partial y^c}{\partial A_{ab}^k}\right)^3 + \epsilon}$$

2) Вес канала k :

$$\omega_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \text{ReLU}\left(\frac{\partial y^c}{\partial A_{ij}^k}\right)$$

3) Итоговая CAM-карта:

$$L^c = \text{ReLU}\left(\sum_k \omega_k^c A^k\right)$$

После построения CAM хуки снимаются, а режим модели восстанавливается, что предотвращает накопление побочных эффектов при многократных вызовах.

CAM-маски и CAM-erasing

Для использования CAM не только как инструмента визуализации, но и как компонента регуляризации реализована функция `_cam_mask_for_batch(model, xb, q=0.80)`, которая строит бинарную маску наиболее важных областей изображения:

- строится CAM для выбранного примера (из батча) и апсемплится до размера входа (H, W);
- выбирается порог по квантилю q ($q=0.8$ соответствует выделению верхних 20% наиболее активных пикселей);
- формируется маска $mask \in \{0,1\}$ размера (B,1,H,W).

Эта маска используется в стратегии CAM-erasing: наиболее дискриминативные области изображения «стираются» (обнуляются) и модель

обязывается искать дополнительные информативные признаки, что снижает риск опоры на узкие или спуриозные корреляции.

Визуализация CAM и контроль локализации

Для качественного анализа реализована функция save_cam_samples(model, xb, out_dir, max_save=4, thr_mask=0.5, area_tag="AREA", step_tag="EPOCH"), которая сохраняет:

- изображение с наложенной тепловой картой CAM (overlay);
- прямоугольник (bounding box), полученный по порогу thr_mask, как оценку локализации активной зоны.

Дополнительно введена количественная проверка «согласованности локализации» - метрика localization faithfulness (localization_faithfulness(model, dl_val, q=0.8, max_batches=30)).

```
for i in range(B):
    x1 = xb[i:i+1]
    cam_map, _ = get_cam_auto(model, x1,
                               layer_paths=("features.denseblock4",
                                             "features.denseblock3"),
                               tta_hflip=True,
                               relu=False, border_crop=0.0)

    cam_up = torch.tensor(cam_map, device=xb.device,
                          dtype=xb.dtype)[None, None]
    cam_up = F.interpolate(cam_up, size=(H, W),
                           mode="bilinear", align_corners=False)[0, 0]
    thr = torch.quantile(cam_up.flatten(), q)
    mask = (cam_up >= thr).to(xb.dtype)[None, None]
    x_erased = x1 * (1 - mask)
    p_full_i = p_full[i].item()
    p_erased_i =
    torch.sigmoid(model(x_erased).squeeze(1)).item()
    drop = max(0.0, p_full_i - p_erased_i)
    drops.append(drop)
    seen += 1
```

Она измеряет среднее падение вероятности положительного класса при стирании области, выделенной CAM: если CAM действительно отражает причинно значимые регионы, то после erasing уверенность модели должна уменьшаться.

CAM для гибридной модели (DenseNet + Swin)

Для гибридной архитектуры реализовано построение двойных CAM-карт compute_and_save_dual_cams(model,inputs,out_logits,save_dir,base_fname_prefix="image"): отдельно для признаков DenseNet-ветки и трансформерной (Swin) ветки. Для этого модель сохраняет активации и градиенты во время прямого прохода, после чего для каждого компонента строится CAM, аппендируется до размера изображения и сохраняется в виде overlay и карты.

```
# CAM для DenseNet
    if d_feat is not None and d_feat.grad is not None:
        d_act_i = d_feat[i].unsqueeze(0).detach()
        d_grad_i = d_feat.grad[i].unsqueeze(0)
        d_cam =
compute_gradcam_from_activation_and_grads(d_act_i, d_grad_i)
        d_cam_up = upsample_cam_to_image(d_cam, H, W)

# CAM для Swin
    if s_feat is not None and s_feat.grad is not None:
        s_act_i = s_feat[i].unsqueeze(0).detach()
        s_grad_i = s_feat.grad[i].unsqueeze(0)
        s_cam =
compute_gradcam_from_activation_and_grads(s_act_i, s_grad_i)
        s_cam_up = upsample_cam_to_image(s_cam, H, W)
```

Это позволяет анализировать участие каждого компонента гибридной модели и сравнивать, какие области изображения считаются важными различными механизмами представления.

Методы и подходы к обучению моделей

В рамках работы исследуются несколько вариантов построения классификаторов для выявления патологий на рентгенограммах MURA. Общая логика экспериментов построена как последовательное усложнение базового решения: от простой baseline-модели к улучшенным схемам обучения и архитектурным модификациям.

Все модели решают задачу бинарной классификации «норма/патология» и обучаются в постановке per-area: для каждой анатомической области формируются отдельные обучающие и валидационные выборки и строится отдельная модель. Оценка качества проводится по метрике ROC AUC, а также по дополнительным метрикам классификации; ключевым является сравнение как на уровне изображений (image-level), так и на уровне исследований (study-level), где предсказания по нескольким снимкам одного исследования агрегируются. Такая методология оценки позволяет более корректно и логически правильно интерпретировать результаты в контексте исходной организации данных MURA.

Далее приводится описание базовой модели (Подход 1) и последующих модификаций (Подходы 2-4), направленных на повышение обобщающей способности, устойчивости к вариативности данных и интерпретируемости решений.

Подход 1 - Baseline DenseNet

В качестве базового уровня качества используется модель DenseNet-121 с предобученными весами ImageNet и заменой классификационной головы на один выходной логит (см. `build_model()`), что соответствует бинарной постановке «норма/патология». Baseline предназначен для получения опорных метрик и дальнейшего сравнения вклада улучшений, вводимых в подходах 2-4.

Схема обучения

Обучение выполняется стандартным стохастическим градиентным спуском в мини-батчах (`batch size = 16`) с использованием оптимизатора Adam и постоянной

скорости обучения LR = 1e-4. Для каждого анатомического раздела XR_* формируются независимые загрузчики данных make_loaders_for_area(area), после чего модель обучается на соответствующей области в течение фиксированного числа эпох (EPOCHS = 5). Такой дизайн обеспечивает сопоставимость результатов между областями при неизменных настройках обучения.

```
def train_one_epoch(model, loader, optimizer, criterion, desc="train"):
    model.train()
    losses = []
    |
    pbar = tqdm(loader, desc=desc, leave=False)
    for x, y in pbar:
        x = x.to(DEVICE)
        y = y.to(DEVICE).float().view(-1, 1) # (B,) -> (B, 1)

        optimizer.zero_grad()
        logits = model(x)
        loss = criterion(logits, y)
        loss.backward()
        optimizer.step()

        lval = loss.item()
        losses.append(lval)

        pbar.set_postfix(loss=np.mean(losses[-50:]))

    return float(np.mean(losses)) if losses else math.nan
```

Рисунок 10 - один проход по обучающей выборке

Функция train_one_epoch() реализует один проход по обучающей выборке: метки приводятся к формату (B, 1), далее вычисляются логиты, значение функции потерь, градиенты и обновление параметров. Для удобства мониторинга в процессе обучения отображается усредненное значение лосса по последним батчам.

Функция потерь и учёт дисбаланса классов

Для baseline используется BCEWithLogitsLoss, объединяющая сигмоидную активацию и бинарную кросс-энтропию в численно устойчивой форме.

```
criterion = nn.BCEWithLogitsLoss(pos_weight=pos_weight)
```

Поскольку в MURA доли классов могут быть неравными, вводится корректировка вклада положительного класса через параметр `pos_weight`, рассчитываемый как отношение числа отрицательных примеров к числу положительных ($\text{pos_weight} = \text{n_neg} / \text{n_pos}$).

```
pos_weight = torch.tensor([(counts.get(0,0) / max(1, counts.get(1,0)))]))
```

Это позволяет сильнее штрафовать ошибки на редком классе `abnormal` и уменьшать смещение модели в сторону доминирующего класса.

Оценка качества (image-level)

Оценка baseline-модели выполняется функцией `evaluate()` на валидационной выборке и проводится на уровне изображений (image-level).

```
all_logits.append(logits.sigmoid().cpu().numpy())
```

Для каждого изображения вычисляется вероятность положительного класса $p = \text{sigmoid}(\text{logit})$, после чего считаются:

- loss (с тем же `BCEWithLogitsLoss`),
- accuracy при пороге 0.5,
- ROC AUC как основная метрика ранжирования.

При невозможности вычисления ROC AUC (например, если во валидации присутствует только один класс) значение метрики помечается как `NaN`, и такой случай не используется для обновления лучшего чекпоинта.

Сохранение чекпоинтов и критерий «лучшей» модели

Для каждой анатомической области сохраняется отдельный лучший чекпойнт по метрике ROC AUC:

```
CKPT_DIR/{area}_best.pt.
```

Чекпоинт содержит веса модели, идентификатор области и размер входного изображения. Сохранение выполняется только при улучшении va_auc относительно текущего best_auc.

Дополнительное дообучение (resume)

Предусмотрен сценарий продолжения обучения (см. Рисунок 11) из ранее сохраненного лучшего чекпоинта.

```
Train: 9752 | Valid: 659 | balance: Counter({0: 5765, 1: 3987})  
/usr/local/lib/python3.12/dist-packages/torch/utils/data/dataloader.py:126: FutureWarning: `torch.cuda.  
warnings.warn(warn_msg)  
XR_WRIST | resume-eval: 0% | 0/42 [00:00<?, ?it/s]  
→ resume XR_WRIST: start_epoch=5, base_auc=0.8946  
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.  
with torch.cuda.amp.autocast(enabled=False):  
XR_WRIST | valid E05: 0% | 0/42 [00:00<?, ?it/s]  
E05 | tr=0.5196 | va=0.4247 | acc=0.8285 | auc=0.8765
```

Рисунок 11 - пример выхода данных при дообучении

При наличии файла {area}_best.pt веса загружаются в модель при условии совпадения метаданных (area, img_size), далее выполняется повторная оценка на валидации для восстановления текущего best_auc и продолжение обучения с заданной эпохи. Важный аспект - при resume сравнение ведется с учетом уже достигнутого лучшего значения AUC, что предотвращает ухудшение качества при перезапуске и сохраняет корректность отбора лучшего состояния модели.

Подход 2 - DenseNet с улучшениями

В данном подходе сохраняется базовая архитектура DenseNet-121, но существенно усложняется процедура обучения: добавляются регуляризаторы на уровне данных (MixUp/CutMix), регуляризация внимания модели (CAM-erasing) и выполняется автоматический подбор гиперпараметров с помощью Optuna⁷. Цель - повысить обобщающую способность модели и стабильность качества в разрезе анатомических областей.

⁷ Optuna - программный фреймворк для автоматической гиперпараметрической оптимизации (Hyperparameter Optimization, HPO), поддерживающий байесовские стратегии поиска (в т.ч. TPE) и раннее прекращение (pruning) неудачных запусков.

Продвинутые аугментации

Помимо стандартных геометрических и фотометрических преобразований (кадрирование, повороты, отражение, небольшие изменения яркости/контраста) во втором подходе применяются аугментации уровня выборки (sample-mixing), которые формируют новые обучающие примеры путем комбинирования изображений и меток внутри батча. В отличие от «классических» аугментаций, меняющих одно изображение, MixUp и CutMix модифицируют распределение обучающих данных и действуют как регуляризаторы, уменьшая переобучение и повышая устойчивость модели к шуму разметки и локальным артефактам.

MixUp выполняет линейное смещивание двух изображений x_i и x_j и их меток y_i, y_j .

```
def mixup_batch(x, y, alpha=0.4):
    if alpha <= 0:
        return x, y, None, None, 1.0

    lam = np.random.beta(alpha, alpha)
    idx = torch.randperm(x.size(0), device=x.device)
    x2 = lam * x + (1. - lam) * x[idx]
    y2 = lam * y + (1. - lam) * y[idx]
    return x2, y2, idx, None, lam |
```

Рисунок 12

Для каждого батча генерируется коэффициент λ из бета-распределения:

$$\lambda \sim Beta(a, a),$$

после чего строятся новые вход и метка:

$$x' = \lambda x_i + (1 - \lambda)x_j, \quad y' = \lambda y_i + (1 - \lambda)y_j$$

В реализации используется случайная перестановка индексов

```
idx = torch.randperm(B),
```

что задает соответствие пар, $(i, idx[i])$ внутри батча. Параметр `mixup_alpha` управляет интенсивностью смещивания: при малых значениях a коэффициент λ

чаще оказывается близким к 0 или 1 (смешивание слабее), при больших - ближе к 0.5 (смешивание сильнее).

С точки зрения обучения MixUp:

- сглаживает границы между классами;
- уменьшает склонность модели к «запоминанию» отдельных примеров;
- повышает устойчивость к случайным шумам и локальным особенностям снимков.

CutMix комбинирует изображения иначе: вместо линейной смеси пикселей в одно изображение вставляется прямоугольный фрагмент из другого.

```
def cutmix_batch(x, y, alpha=1.0):
    if alpha <= 0:
        return x, y, None, None, 1.0
    lam = np.random.beta(alpha, alpha)
    B, C, H, W = x.size()
    idx = torch.randperm(B, device=x.device)
    x1, y1, x2, y2 = _rand_bbox(W, H, lam)
    x2img = x[idx]
    x_aug = x.clone()

    x_aug[:, :, y1:y2, x1:x2] = x2img[:, :, y1:y2, x1:x2]
    box_area = (x2 - x1) * (y2 - y1)
    lam_eff = 1.0 - box_area / float(W * H)
    y2 = lam_eff * y + (1. - lam_eff) * y[idx]
    return x_aug, y2, idx, (x1, y1, x2, y2), lam_eff
```

Рисунок 13

Коэффициент λ также берется из $Beta(a, a)$, но интерпретируется как доля исходного изображения, которая должна сохраниться. На его основе вычисляется относительная площадь вставки:

$$area_{patch} = 1 - \lambda$$

Далее функция `_rand_bbox(W,H,lambda)` формирует координаты прямоугольника: случайно выбирается центр (c_x, c_y) и вычисляются размеры прямоугольника пропорционально $\sqrt{1 - \lambda}$, после чего координаты ограничиваются границами изображения. Вставка выполняется одинаковыми координатами для всех элементов

батча, но «донор» патча для каждого элемента выбирается через перестановку idx . Ключевой момент - корректировка метки по фактической площади вставленного фрагмента, так как из-за обрезки прямоугольника на границах реальная площадь может отличаться от заданной $(1 - \lambda)$. В коде вычисляется эффективный коэффициент:

$$\lambda_{eff} = 1 - \frac{S_{patch}}{W * H}$$

и метка смещивается как:

$$y' = \lambda_{eff} y_i + (1 - \lambda_{eff}) y_j$$

Благодаря этому модель получает согласованные пары «изображение-метка». С точки зрения обучения CutMix:

- заставляет модель использовать контекст и более распределенные признаки;
- работает как регуляризация локализации (модель видит частично «зашумленные» изображения и учится быть устойчивой);
- дает буст на данных, где важные признаки локальны и могут занимать небольшую часть кадра

В данном пайплайне MixUp и CutMix применяются на уровне батча с вероятностью `aug_p`. Режим `mix_mode="auto"` означает, что при срабатывании аугментации случайно выбирается один из двух методов (`mixup` или `cutmix`). Это позволяет избежать «переобучения» на одной конкретной схеме смешивания и дает более разнообразное регуляризующее воздействие. Если `mixup_alpha` или `cutmix_alpha` установлены в 0, соответствующая аугментация фактически отключается (в Optuna это позволяет автоматически выбирать, нужен ли метод для данной области).

Модифицированный цикл обучения

В Подходе 2 базовый цикл обучения расширен так, чтобы в одном проходе по данным сочетать регуляризацию на уровне данных (MixUp/CutMix) и

регуляризацию внимания (CAM-erasing). Это реализовано в функции `train_one_epoch(model, loader, optimizer, criterion, desc="train", use_cam_erasing=True, erase_every=8, erase_q=0.80, erase_k=1, alpha_cam=0.5, use_mix=True, mix_mode="auto", aug_p=0.5, mixup_alpha=0.4, cutmix_alpha=1.0)`, которая сохраняет стандартную схему

forward → loss → backward → step],

но добавляет два дополнительных механизма.

1) Стохастическое применение MixUp/CutMix к батчу.

Для каждого батча с вероятностью `aug_p` выполняется одно из преобразований:

- MixUp - линейное смешивание двух изображений и меток;
- CutMix - вставка прямоугольного фрагмента одного изображения в другое с коррекцией метки по площади вставки (`lam_eff`).

Режим задается параметром `mix_mode` (фиксированный выбор или `auto`, когда MixUp/CutMix выбираются случайно).

```
if use_mix and random.random() < aug_p:
    mode = mix_mode if mix_mode != "auto" else
random.choice(["mixup", "cutmix"])
    if mode == "mixup":
        x_in, y_in, _, _, _ = mixup_batch(xb, yb,
alpha=mixup_alpha)
    else:
        x_in, y_in, _, _, _ = cutmix_batch(xb, yb,
alpha=cutmix_alpha)
```

Интенсивность смешивания регулируется параметрами `mixup_alpha` и `cutmix_alpha` (параметры Beta-распределения для коэффициента смешивания).

2) CAM-erasing как дополнительная ветка потерь.

Каждые `erase_every` батчей вычисляется CAM-маска по текущей модели: из карты важности выделяются наиболее активные пиксели (по квантилю `erase_q`, например топ-20%), затем на их месте выполняется «стирание»

```
xb_erased = xb * (1 - mask).
```

После этого модель прогоняется второй раз на xb_erased, вычисляется дополнительная потеря loss2, и итоговая функция потеря берется как взвешенная смесь:

$$loss = (1 - \alpha_{cam}) \cdot loss + \alpha_{cam} \cdot loss2,$$

где alpha_cam задает вклад обучения на «стертых» изображениях.

```
if use_cam_erasing and (step % erase_every == 0):  
  
    mask = _cam_mask_for_batch(model, xb,  
k_samples=erase_k, q=erase_q).detach()  
    xb_erased = xb * (1 - mask)  
    logits2 = model(xb_erased).squeeze(1)  
    loss2 = criterion(logits2, yb.squeeze(1))  
    loss = (1.0 - alpha_cam) * loss + alpha_cam *  
loss2
```

Почему CAM-erasing считается на оригинальном батче?

Маска строится по исходному xb, а не по MixUp/CutMix-версии, чтобы карта внимания соответствовала реальным структурам изображения и неискажалась искусственным смешиванием! В результате модифицированный цикл обучения одновременно снижает переобучение за счет смешивания примеров и сглаживания границ классов, и уменьшает зависимость модели от одной «самой яркой» дискриминативной области, стимулируя использование альтернативных признаков, что особенно полезно, учитывая наличие надписей на снимках.

Гиперпараметрическая оптимизация

Гиперпараметры существенно влияют на итоговое качество и по-разному работают в разных анатомических областях MURA. Поэтому в подходе 2 используется автоматизированная гиперпараметрическая оптимизация вместо ручного подбора.

Optuna - это открытый фреймворк для НРО, который организует поиск по пространству гиперпараметров в виде серии независимых запусков (trials), фиксирует результаты и поддерживает механизмы ускорения поиска за счет интеллектуального сэмплирования и ранней остановки слабых конфигураций.

Как это устроено в работе (коротко по коду):

- Оптимизация проводится per-area, т.е. отдельно для каждой области XR_*, чтобы учесть различия в данных и сложности задач.
- Целевая метрика - study-level ROC-AUC на валидации (evaluate_study), поскольку в MURA разметка задана на уровне исследования, а не отдельного кадра.
- Для каждой области выполняется N_TRIALS = 25 попыток, каждая попытка обучается коротко (EPOCHS_TUNE = 2) - это компромисс между качеством отбора и вычислительной стоимостью.

```
def objective(trial: optuna.trial.Trial):  
    lr          = trial.suggest_loguniform("lr", 1e-5, 3e-4)  
    weight_decay= trial.suggest_loguniform("weight_decay",  
1e-7, 1e-3)  
    mixup_alpha = trial.suggest_float("mixup_alpha", 0.0,  
0.6)  
    cutmix_alpha= trial.suggest_float("cutmix_alpha", 0.0,  
1.5)  
    aug_p       = trial.suggest_float("aug_p", 0.3, 0.8)  
    erase_q     = trial.suggest_float("erase_q", 0.70, 0.90)  
    alpha_cam   = trial.suggest_float("alpha_cam", 0.3, 0.7)  
    erase_every = trial.suggest_int("erase_every", 6, 12)
```

Сэмплер и прунинг:

- В Optuna используется TPESampler - байесовская стратегия на основе Tree-structured Parzen Estimator (TPE), которая обычно эффективнее полного перебора/грида, когда гиперпараметров много и вычислительный ресурс ограничен.

Идея такая:

- Optuna накапливает результаты уже запущенных trial.
- Делит прошлые trial на «хорошие» и «плохие» (например, верхние X% по AUC).
- Строит две вероятностные модели распределений параметров:
 - $l(x) = p(x|y \text{ хорошее})$
 - $g(x) = p(x|y \text{ плохое})$
- Затем предлагает новые параметры там, где отношение $l(x)/g(x)$ максимально
 - то есть похоже на хорошие и непохоже на плохие.
 - Для ускорения применяется MedianPruner⁸: trial может быть прерван досрочно, если его промежуточный результат хуже медианного уровня уже выполненных trial на сопоставимой глубине обучения. Это экономит время, не тратя эпохи на заведомо слабые конфигурации.

Как работает:

- В каждом trial после каждой эпохи делается `trial.report(auc_st, ep)`.
- Optuna смотрит: «на этой же эпохе у уже завершенных trial какая медианная метрика?»
- Если текущий trial хуже медианы, и выполнены условия прунинга - он прерывается (`TrialPruned`).

```
if trial.should_prune():
    raise optuna.exceptions.TrialPruned()
```

Если после первой эпохи trial плохой, его обучение прерывается без траты времени на вторую эпоху.

```
pruner=MedianPruner(n_warmup_steps=1)
```

Для каждой области поддерживается глобальный лучший результат `best_area["auc"]`; при улучшении сохраняется чекпоинт

```
CKPT_DIR/{area}_best_optuna_studyID.pt
```

⁸ Это ранняя остановка trial на основе промежуточных метрик.

вместе с best_auc_study и найденными hpo_params. Устанавливается SEED, чтобы частично стабилизировать случайность (NumPy/PyTorch) и сделать сравнение trial более корректным.

```
== Возобновление HPO для XR_ELBOW ==
Downloading: "https://download.pytorch.org/models/densenet121-a639ec97.pth" to /root/.cache/torch/hub/checkpoints/densenet121-a639ec97.pth
100%|██████████| 30.8M/30.8M [00:00<00:00, 189MB/s]
XR_ELBOW | HPO tr E01: 0% | 0/309 [00:00<?, ?it/s]
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast` with torch.cuda.amp.autocast(enabled=False):
XR_ELBOW | HPO va (study) E01: 0% | 0/30 [00:00<?, ?it/s]
XR_ELBOW | HPO tr E02: 0% | 0/309 [00:00<?, ?it/s]
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast` with torch.cuda.amp.autocast(enabled=False):
XR_ELBOW | HPO va (study) E02: 0% | 0/30 [00:00<?, ?it/s]
XR_ELBOW | HPO tr E01: 0% | 0/309 [00:00<?, ?it/s]
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast` with torch.cuda.amp.autocast(enabled=False):
XR_ELBOW | HPO va (study) E01: 0% | 0/30 [00:00<?, ?it/s]
XR_ELBOW | HPO tr E01: 0% | 0/309 [00:00<?, ?it/s]
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast` with torch.cuda.amp.autocast(enabled=False):
XR_ELBOW | HPO va (study) E01: 0% | 0/30 [00:00<?, ?it/s]
XR_ELBOW | HPO tr E01: 0% | 0/309 [00:00<?, ?it/s]
/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast` with torch.cuda.amp.autocast(enabled=False):
XR_ELBOW | HPO va (study) E01: 0% | 0/30 [00:00<?, ?it/s]
XR_ELBOW | HPO tr E01: 0% | 0/309 [00:00<?, ?it/s]
```

Рисунок 14 - Возобновление обучения optuna

Optuna позволяет системно и воспроизводимо подобрать комбинацию гиперпараметров, которая максимизирует study-level AUC для каждой области, вместо ручного тюнинга, который обычно неустойчив и плохо переносится между областями.

Подход 3 - Финальная DenseNet (двуухфазное обучение)

После подбора гиперпараметров (Подход 2) выполняется финальный прогон обучения, цель которого - получить максимально стабильную и воспроизводимую модель для каждой анатомической области. Подход 3 сочетает: (1) использование найденных Optuna-гиперпараметров как стартовой конфигурации, (2) длительное обучение (EPOCHS_FINAL=50), (3) двухфазный режим fine-tuning, и (4) аккуратную систему логирования и чекпоинтов.

Для каждой области сохраняются два типа чекпоинтов:

- "{CKPT_DIR}/{area}_FINAL_best.pt" - модель с лучшим значением study-level AUC на валидации;

- "{CKPT_DIR}/{area}_FINAL_last.pt" - состояние последней эпохи для автоматического возобновления обучения.

Также сохраняются журналы обучения:

- "{CKPT_DIR}/{area}_FINAL_history.csv" - история метрик по эпохам для конкретной области;
- final_training_all_epochs_history.csv - единая история по всем областям;
- final_training_summary.csv - итоговая сводная таблица по областям, собранная из *_FINAL_best.pt.

Такой метод был выбран, чтобы: (а) восстанавливать обучение после перерывов, (б) анализировать динамику метрик, и (в) прозрачно фиксировать лучшую модель, выбранную по целевому критерию.

Перед финальным обучением для каждой области из файла optuna_summary_from_ckpts.csv извлекаются лучшие параметры (см. Рисунок 15). Это говорит о том, что финальный прогон стартует не с универсальных настроек, а с конфигурации, уже показавшей высокий результат на данной области.

```
row = df_params.loc[area]
base_lr      = float(row["lr"])
weight_decay = float(row["weight_decay"])
mixup_alpha  = float(row["mixup_alpha"])
cutmix_alpha = float(row["cutmix_alpha"])
aug_p        = float(row["aug_p"])
erase_q      = float(row["erase_q"])
alpha_cam    = float(row["alpha_cam"])
erase_every  = int(row["erase_every"])
```

Рисунок 15 - Гиперпараметры optuna

Обучение разделено на две фазы:

```

if epoch < PHASE2_START_EPOCH:
    phase = 1
    current_mixup_alpha = mixup_alpha
    current_cutmix_alpha = cutmix_alpha
    current_aug_p = aug_p
    use_mix = True
    for pg in optimizer.param_groups:
        pg["lr"] = base_lr
else:
    phase = 2
    for pg in optimizer.param_groups:
        pg["lr"] = base_lr * LR_DECAY_PHASE2
    current_mixup_alpha = 0.0
    current_cutmix_alpha = 0.0
    current_aug_p = 0.0
    use_mix = False
use_cam = (epoch >= CAM_ERASE_START_EPOCH)
tr_loss = train_one_epoch(
    model, dl_tr, optimizer, criterion,
    desc=f"{area} | FINAL (phase {phase}) tr E{epoch:02d}",
    use_cam_erasing=use_cam,
    erase_every=erase_every,
    erase_q=erase_q,
    alpha_cam=alpha_cam,
    use_mix=use_mix,
    mix_mode="auto",
    aug_p=current_aug_p,
    mixup_alpha=current_mixup_alpha,
    cutmix_alpha=current_cutmix_alpha,

```

Рисунок 16 (а) - Включение CAM-erasing

Рисунок 16 (б) - Двухфазное обучение

Фаза 1 (до PHASE2_START_EPOCH, по умолчанию до 15-й эпохи, см. Рисунок 16 (а)):

- используется базовый learning rate base_lr, найденный Optuna;
- активны MixUp/CutMix с вероятностью aug_p;
- CAM-erasing включается не сразу (см. Рисунок 16 (б)), а с CAM_ERASE_START_EPOCH (с 3-й эпохи), чтобы не ломать раннюю сходимость и дать модели сформировать базовые признаки.

Фаза 2 (с PHASE2_START_EPOCH, начиная с 16-й эпохи, см. Рисунок 16 (а)):

- выполняется тонкая доводка (fine-tune): learning rate уменьшается в LR_DECAY_PHASE2 раз ($lr = base_lr * 0.1$);
- MixUp/CutMix отключаются ($\alpha=0$, $aug_p=0$), чтобы не добавлять шум в момент, когда модель уточняет решение и калибрует границы классов.

Сначала модель учится устойчивым признакам на более шумном и разнообразном распределении (регуляризация), затем донастраивается на реальных данных без сильных искажений.

На каждой эпохе считается полный набор метрик как:

- image-level: val_loss_img, val_acc_img, val_auc_img;
- study-level: val_acc_study, val_auc_study, val_f1_study.

```

(va_loss_img, va_acc_img, va_auc_img,
va_acc_st, va_auc_st, va_f1_st) = evaluate_study(
    model, dl_va, desc=f"{{area}} | FINAL (phase {{phase}}) va (study) E{{epoch:02d}}"
)

dt = time.time() - t0
current_lr = optimizer.param_groups[0]["lr"]

```

Рисунок 17

При этом оптимизация выбора лучшего состояния ведется строго по `val_auc_study`, поскольку эта метрика наиболее релевантна постановке задачи. При улучшении `val_auc_study` чекпоинт `*_FINAL_best.pt` перезаписывается, и фиксируется номер лучшей эпохи (`best_epoch`) и соответствующие метрики.

Перед началом прогона реализованы два механизма восстановления:

1. если уже существует `*_FINAL_best.pt`, из него извлекается предыдущее лучшее значение `best_auc_study`, чтобы продолжение сравнения шло с учетом старого лучшего;
2. если существует `*_FINAL_last.pt`, модель и (при наличии) оптимизатор загружаются и обучение продолжается с `epoch = last_epoch + 1`.

Это делает финальное обучение устойчивым к прерываниям и предотвращает потерю достигнутого максимума по целевой метрике.

После завершения финального обучения автоматически собирается глобальная таблица результатов: для всех найденных файлов `*_FINAL_best.pt` извлекаются метрики и формируется `final_training_summary.csv` (с сортировкой по AUC). Это обеспечивает единый формат отчетности по областям и упрощает последующий сравнительный анализ подходов.

```

===== FINAL TRAINING: XR_WRIST =====
Train images: 9752 | Valid images: 659 | class balance (train): Counter({0: 5765, 1: 3987})
Использование гиперпараметров (фаза 1):
    lr=0.000063, weight_decay=1.19e-04
    mixup_alpha=0.121, cutmix_alpha=0.151, aug_p=0.512
    erase_q=0.701, alpha_cam=0.624, erase_every=10
    CAM-erasing включится с эпохи >= 3
    В фазе 2 (с эпохи 16) lr будет умножен на 0.1, MixUp/CutMix будут отключены.
Downloading: "https://download.pytorch.org/models/densenet121-a639ec97.pth" to /root/.cache/torch/hub/checkpoints/densenet121-a639ec97.pth
100% [██████████] 30.8M/30.8M [00:00<00:00, 189MB/s]
Найден best чекпоинт: AUC_study=0.9214 (epoch=3)
Найден чекпоинт последней эпохи: /content/drive/MyDrive/курсовая mura/mura_ckpts/XR_WRIST_FINAL_last.pt
Возобновление обучения с эпохи 49

/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
with torch.cuda.amp.autocast(enabled=False):

E49 [phase 2] | lr=6.267355e-06 | tr_loss=0.0268 | img: loss=0.9010 acc=0.8300 auc=0.8719 | study: acc=0.8608 auc=0.9001 f1=0.8177 | 167.1s

/tmp/ipython-input-1150745146.py:126: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
with torch.cuda.amp.autocast(enabled=False):

E50 [phase 2] | lr=6.267355e-06 | tr_loss=0.0233 | img: loss=0.9049 acc=0.8376 auc=0.8737 | study: acc=0.8734 auc=0.9028 f1=0.8315 | 166.25
История по эпохам для XR_WRIST сохранена в /content/drive/MyDrive/курсовая mura/mura_ckpts/XR_WRIST_FINAL_history.csv

area epoch phase train_val loss_val loss_val_img val_auc_img val_auc_study val_auc_study_val_ft_study cam_erasing_on lr_effective base_lr weight_decay mixup_alpha_base cutmix_alpha_base aug_p_base erase_q_base alpha_cam erase_every cam_erase_start_epoch phase2_start_epoch lr_decay_phase2 best_epoch
XR_WRIST 3 1 0.51625 0.37339 0.04522 0.096029 0.089198 0.923429 0.038601 0.000003 0.000003 0.000119 0.120894 0.151029 0.512462 0.760714 0.623986 10 3 16 0.1 47
Saved full epoch-by-epoch history (all runs merged) -> /content/drive/MyDrive/ipycode/mura/mura_ckpts/final_training_all_epochs.history.csv

==== GLOBAL FINAL TRAIN-LEVEL RESULTS (from all 1 EPOCHS history) ====
area epoch phase train_val loss_val loss_val_img val_auc_img val_auc_study val_auc_study_val_ft_study cam_erasing_on lr_effective base_lr weight_decay mixup_alpha_base cutmix_alpha_base aug_p_base erase_q_base alpha_cam erase_every cam_erase_start_epoch phase2_start_epoch lr_decay_phase2 best_epoch
XR_EMERIS 47 2 0.830255 0.498551 0.064543 0.919983 0.874070 0.933375 0.0723280 0.000009 0.000009 0.244134 0.299894 0.566209 0.820812 0.671830 6 3 16 0.1 47
XR_EYE 1 1 0.51625 0.37339 0.04522 0.096029 0.089198 0.923429 0.038601 0.000003 0.000003 0.000119 0.120894 0.151029 0.512462 0.760714 0.623986 10 3 16 0.1 47
XR_EAR 9 1 0.446812 0.372568 0.078797 0.912967 0.945430 0.918149 0.0327668 0.000008 0.000008 0.651522e-04 0.439196 0.870899 0.751199 0.323233 12 3 16 0.1 47
XR_FOREARM 15 1 0.450774 0.412396 0.081101 0.893716 0.892126 0.890862 0.0326468 0.000013 0.000013 0.000002 0.206319 0.610858 0.740945 0.801239 0.391945 9 3 16 0.1 47
XR_HAND 6 1 0.445564 0.405564 0.080001 0.892739 0.895949 0.889514 0.0327668 0.000017 0.000017 0.000002 0.116072 1.181334 0.657955 0.831654 0.445980 10 3 16 0.1 47
XR_SHOULDER 6 1 0.451763 0.450564 0.086767 0.899499 0.895928 0.889909 0.0327668 0.000107 0.000107 0.000002 0.116072 1.181334 0.657955 0.831654 0.445980 9 3 16 0.1 47
XR_WIND 1 1 0.345585 0.522463 0.763843 0.853374 0.766431 0.862286 0.066607 0.000093 0.000093 1.228199e-07 0.493478 0.017358 0.250902 0.873799 0.354872 11 3 16 0.1 47

```

Рисунок 18 - Завершение финального обучения и глобальная таблица результатов

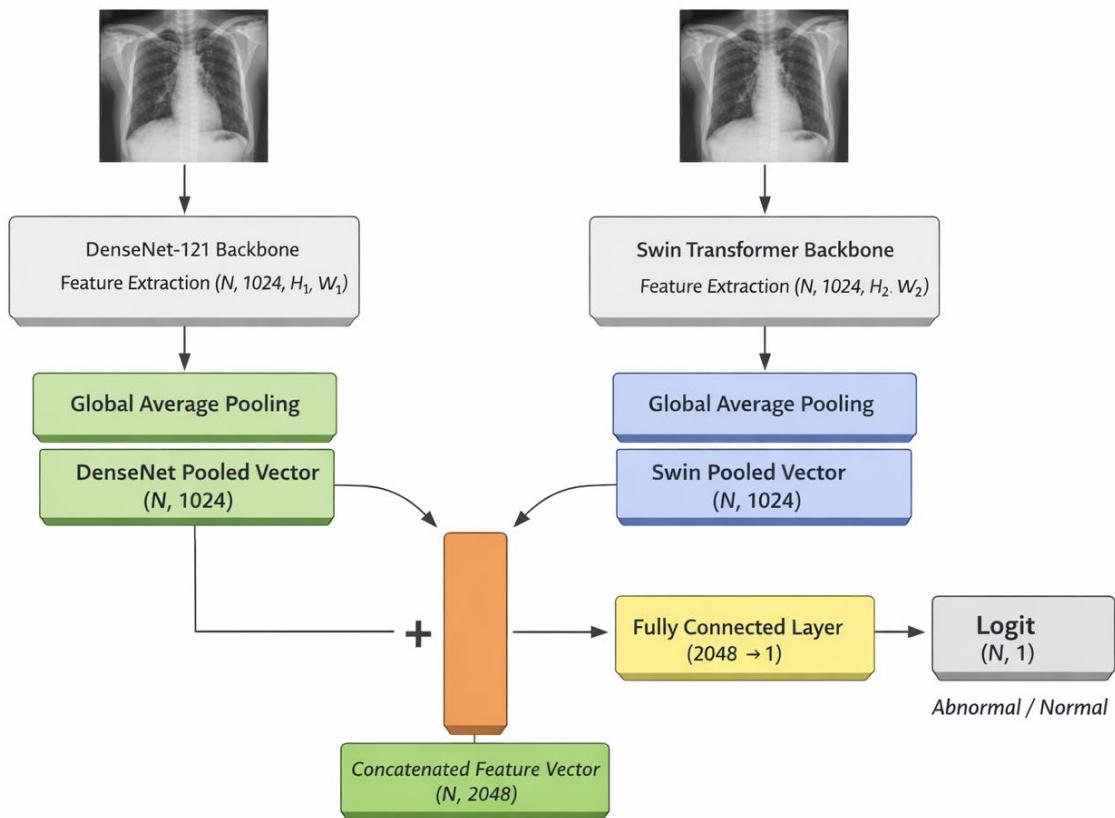
Подход 4 - Гибридная модель DenseNet + Swin Transformer

Подход 4 вводит гибридную архитектуру, сочетающую сверточные признаки DenseNet и трансформерные признаки Swin Transformer в единой модели. Гибридный подход (DenseNet + Swin) был выбран и протестирован в формате гипотезы: предполагалось, что для наиболее сложных анатомических областей (XR_SHOULDER, XR_FINGER, XR_HAND) стандартная CNN-модель может опираться на ограниченный набор локальных признаков и демонстрировать нестабильность обобщения. В качестве альтернативы была выдвинута гипотеза, что добавление трансформерного компонента Swin Transformer, способного эффективнее агрегировать контекст и учитывать связи между удаленными участками изображения, позволит усилить представления и повысить качество классификации именно на трудных случаях. При этом гибридизация выполнена не как замена базовой DenseNet, а как объединение признаков двух энкодеров (feature-level fusion). Такая постановка позволит проверить гипотезу о взаимодополняемости CNN и transformer признаков: если Swin действительно добавит полезную информацию, модель выиграет за счет конкатенации признаков, а анализ dual-CAM позволит оценить, различаются ли зоны внимания DenseNet и Swin на одних и тех же изображениях.

Архитектура гибридной модели:

```
# DenseNet121
    self.densenet = timm.create_model(
        'densenet121',
        pretrained=pretrained,
        features_only=True,
        out_indices=(3, )
    )

# Swin Base
    self.swin = timm.create_model(
        'swin_base_patch4_window7_224',
        pretrained=pretrained,
        features_only=True,
        out_indices=(3, )
)
```



HybridModel включает два предобученных энкодера (timm⁹, режим features_only=True¹⁰):

- DenseNet121 → извлекается последний уровень признаков, тензор вида (N, 1024, H1, W1);
- Swin Base (swin_base_patch4_window7_224) → извлекается последний уровень признаков; при необходимости выполняется преобразование формата NHWC → NCHW.

```
if s_feat.ndim == 4 and s_feat.shape[1] != d_feat.shape[1]:  
    s_feat = s_feat.permute(0, 3, 1, 2).contiguous()
```

Далее применяется Global Average Pooling (AdaptiveAvgPool2d(1)), переводящий карты признаков в векторы:

- DenseNet: (N, 1024, H, W) → (N, 1024)
- Swin: (N, 1024, H, W) → (N, 1024)

Векторы конкатенируются в общий признак размера (N, 2048) и подаются в линейную голову:

- FC: 2048 → 1 (один логит для BCEWithLogitsLoss).

Это объединение можно трактовать как «feature-level fusion»: модель учится использовать сильные стороны каждого энкодера, не заставляя один модуль решать задачу в одиночку.

Для гибридного прогона сохраняются:

- {area}_FINAL_trans_last.pt - последний чекпоинт (для возобновления);
- {area}_FINAL_trans_best.pt - лучший чекпоинт по валидационной AUC;
- {area}_hybrid_val_preds.csv - выгрузка предсказаний на валидации.

Обучение разделено на две фазы, чтобы стабильно подключить второй энкодер и не разрушить признаки.

Фаза 1 (основное обучение, EPOCHS_FINAL=50):

⁹ timm (PyTorch Image Models) - библиотека моделей и предобученных весов для PyTorch, предоставляющая единый интерфейс создания архитектур и получения промежуточных признаков

¹⁰ Параметр features_only=True включает режим feature extraction: вместо классификационного выхода модель возвращает набор промежуточных feature maps, пригодных для построения собственных голов (classification head) или объединения признаков в гибридных архитектурах.

- DenseNet полностью заморожен (`requires_grad=False`), обучаются Swin + FC-голова;
- оптимизация Adam, фиксированные гиперпараметры: `HYBRID_LR_PHASE1 = 3e-4`, `HYBRID_WEIGHT_DECAY = 1e-5`;
- дисбаланс классов учитывается через `pos_weight` в `BCEWithLogitsLoss`.

Цель в том, чтобы сначала научить второй канал (Swin) извлекать полезные признаки под задачу и согласовать его с классификационной головой, не трогая DenseNet.

Фаза 2 (тонкая доводка, `FT_EPOCHS=10`):

- частично размораживаются верхние слои DenseNet (`denseblock4` и `norm5`);
- learning rate снижается до `HYBRID_LR_PHASE2 = 1e-5`;
- выполняется аккуратный fine-tune всей связки (DenseNet верх + Swin + FC).

```
for name, p in model.densenet.named_parameters():
    if "denseblock4" in name or "norm5" in name:
        p.requires_grad = True

    ft_lr = HYBRID_LR_PHASE2
    print(f"PHASE2 lr={ft_lr:.6f}, weight_decay={weight_decay:.2e}")

optimizer = optim.Adam(
    filter(lambda p: p.requires_grad, model.parameters()),
    lr=ft_lr,
    weight_decay=weight_decay
)
```

Рисунок 19 - Фаза 2

Такой режим минимизирует риск сломать предобученные признаки DenseNet и позволяет донастроить общую репрезентацию под сложную область.

В смысле гибрида метрика отбора - валидационный AUC на уровне изображений (image-level), вычисляемый по вероятностям `sigmoid(logit)`. Лучший чекпоинт сохраняется при улучшении `val_auc`.

Для интерпретируемости в `HybridModel` предусмотрен режим сохранения активаций и градиентов (`store_activations_for_cam=True`). В определённые эпохи строятся две CAM-карты:

- CAM по признакам DenseNet (`d_feat`);
- CAM по признакам Swin (`s_feat`).

Это позволяет анализировать, который из энкодеров на каких примерах видит патологию и как меняется внимание при fine-tune. CAM сохраняются периодически:

- в фазе 1 - раз в 5 эпох (и на 1-й),
- во 2-й фазе - раз в 2 эпохи, так как изменения более тонкие и интересные для анализа

В подходах 2-3 в качестве основной метрики использовался study-level ROC-AUC, поскольку датасет MURA размечен на уровне исследования, и именно такой протокол оценки принят в базовой работе Stanford и удобен для сопоставления результатов. Поэтому финальная DenseNet оптимизировалась и отбиралась по val_auc_study, а image-level метрики логировались как вспомогательные для анализа поведения модели по отдельным снимкам. В гибридном подходе (DenseNet + Swin) эксперимент носит характер целевого усиления для наиболее сложной области, где внутри одного исследования присутствуют проекции с неодинаковой информативностью: патология может быть отчетливо видна лишь на части снимков, а остальные кадры выглядят близкими к норме. При простой агрегации по исследованию такой сценарий приводит к размытию сигнала и может занижать оценку, тогда как image-level AUC напрямую отражает способность модели выделять патологию на тех кадрах, где она визуализируется. Кроме того, в текущей реализации гибридного обучения выбор лучшего чекпоинта сделан по image-level AUC как по оперативному прокси для мониторинга качества и ускорения эксперимента; при необходимости строгого сопоставления с подходами 2-3 оценка гибрида может быть расширена до study-level (с агрегацией по study_id, предпочтительно не средним, а max/MIL-схемой для указанных областей).

Экспериментальные результаты

Далее представлены количественные результаты и качественный анализ моделей, обученных в per-area постановке на MURA. Лучшие чекпоинты выбирались по ROC-AUC на уровне исследования (study-level), после чего рассчитывались дополнительные пороговые метрики и выполнялся анализ ошибок.

Метрики оценки качества

Оценка качества проводилась раздельно по анатомическим областям (per-area) на валидационной выборке. Для каждого снимка модель выдает логит z и вероятность патологии.

$$p = \sigma(z) = \frac{1}{1 + e^{-z}}$$

Два уровня оценки: image-level и study-level.

Поскольку в MURA одно исследование содержит несколько снимков, метрики считались:

- Image-level - по каждому изображению отдельно (каждый файл = один объект).
- Study-level - по исследованиям: вероятности по всем снимкам одного исследования агрегируются:

$$p_{study} = \frac{1}{K} \sum_{i=1}^K p_i$$

после чего метрики вычисляются уже по исследованиям как по единицам наблюдения. Именно study-level качество использовалось как приоритетное для выбора лучших чекпоинтов, т.к. соответствует клинической постановке норма/патология на уровне исследования.

Основная метрика: ROC-AUC

В качестве ключевой метрики использовалась площадь под ROC-кривой (ROC-AUC):

- не зависит от фиксированного порога,
- корректно сравнивает модели при дисбалансе классов,
- позволяет сопоставлять результаты между областями.

ROC-AUC считалась как для image-level, так и для study-level.

Для практической интерпретации дополнительно использовались метрики при фиксированном пороге $t=0.5$:

- Accuracy: $\frac{TP+TN}{TP+TN+FP+FN}$
- Precision: $\frac{TP}{TP+FP}$
- Recall (Sensitivity): $\frac{TP}{TP+FN}$
- F1-score: $2 * \frac{Precision*Recall}{Precision+Recall}$

Для областей с выраженным дисбалансом дополнительно анализировались:

- PR-кривые (Precision-Recall),
- Average Precision (AP) как интегральная характеристика качества ранжирования по PR-кривой.

Для каждого per-area классификатора строились матрицы ошибок при $t=0.5$, что позволило:

- видеть структуру ошибок (FP/FN),
- сравнивать области,
- выделять трудные случаи (например, области с высоким FN).

Так как вероятности нейросетей часто бывают плохо откалиброваны, дополнительно оценивалась согласованность уверенности модели с фактической точностью:

- Reliability diagram (по бинам уверенности),
- Expected Calibration Error (ECE) - средневзвешенная разница между confidence и accuracy по бинам:

$$ECE = \sum_{m=1}^M \frac{|B_m|}{N} | acc(B_m) - conf(B_m) |$$

Это важно для интерпретации предсказаний как вероятностей, особенно при применении модели как вспомогательного инструмента.

Помимо per-area таблиц рассчитывались агрегированные сводки по областям:

- macro-average (усреднение метрик по областям),
- weighted-average (усреднение с весами по размеру валидации области), чтобы получить обобщенную картину качества по всему набору MURA.

Результаты baseline-модели

	area	val_size	val_loss	AUC	ACC	Precision	Recall	F1
	XR_ELBOW	465	0.3807	0.9126	0.8366	0.9185	0.7348	0.8164
	XR_HUMERUS	288	0.4167	0.9103	0.8160	0.7702	0.8857	0.8239
	XR_FOREARM	301	0.4272	0.9007	0.8272	0.9160	0.7219	0.8074
	XR_WRIST	659	0.3946	0.8946	0.8422	0.8577	0.7763	0.8149
	XR_SHOULDER	563	0.4600	0.8639	0.7993	0.8161	0.7662	0.7904
	XR_FINGER	461	0.5217	0.8451	0.7527	0.8446	0.6599	0.7409
	XR_HAND	460	0.5131	0.8411	0.7283	0.6391	0.7778	0.7017

Рисунок 20 - Результат по областям Подхода 1

В таблице (см. Рисунок 20) приведены метрики качества baseline-модели (DenseNet121) для каждой анатомической области. Следует учитывать, что ROC-AUC отражает прежде всего способность модели ранжировать объекты (насколько типичные «патологические» исследования получают более высокий score по сравнению с «нормой») и не фиксирует конкретную рабочую точку классификатора. Напротив, метрики ACC/Precision/Recall/F1 рассчитываются при заданном пороге принятия решения (в данной работе - 0.5) и характеризуют операционное поведение системы при бинарном решении.

По результатам видно, что для ряда областей (XR_ELBOW, XR_HUMERUS, XR_FOREARM) достигаются относительно высокие значения ROC-AUC, что свидетельствует о хорошем разделении классов по score модели. Однако при фиксированном пороге 0.5 профиль ошибок оказывается неоднородным: для части областей наблюдается повышенная точность при более низкой полноте (когда модель реже выдает положительный класс), тогда как для других областей возможна обратная ситуация - высокая полнота при росте ложноположительных срабатываний (режим более «чувствительного» детектора).

Для областей XR_HAND и XR_FINGER значения ROC-AUC ниже, что согласуется с более сложной структурой данных и, как правило, приводит к большему перекрытию распределений предсказанных вероятностей для классов

«норма/патология». Это отражается в снижении операционных метрик и подтверждается анализом матриц ошибок и распределений вероятностей (см. Рисунок 21). Область XR_WRIST демонстрирует более сбалансированное сочетание: высокая ранжирующая способность модели (ROC-AUC) сопровождается относительно стабильными значениями метрик при пороге 0.5. Это указывает на более выраженное разделение классов и, как следствие, более устойчивую рабочую точку классификатора по сравнению с наиболее трудными областями.

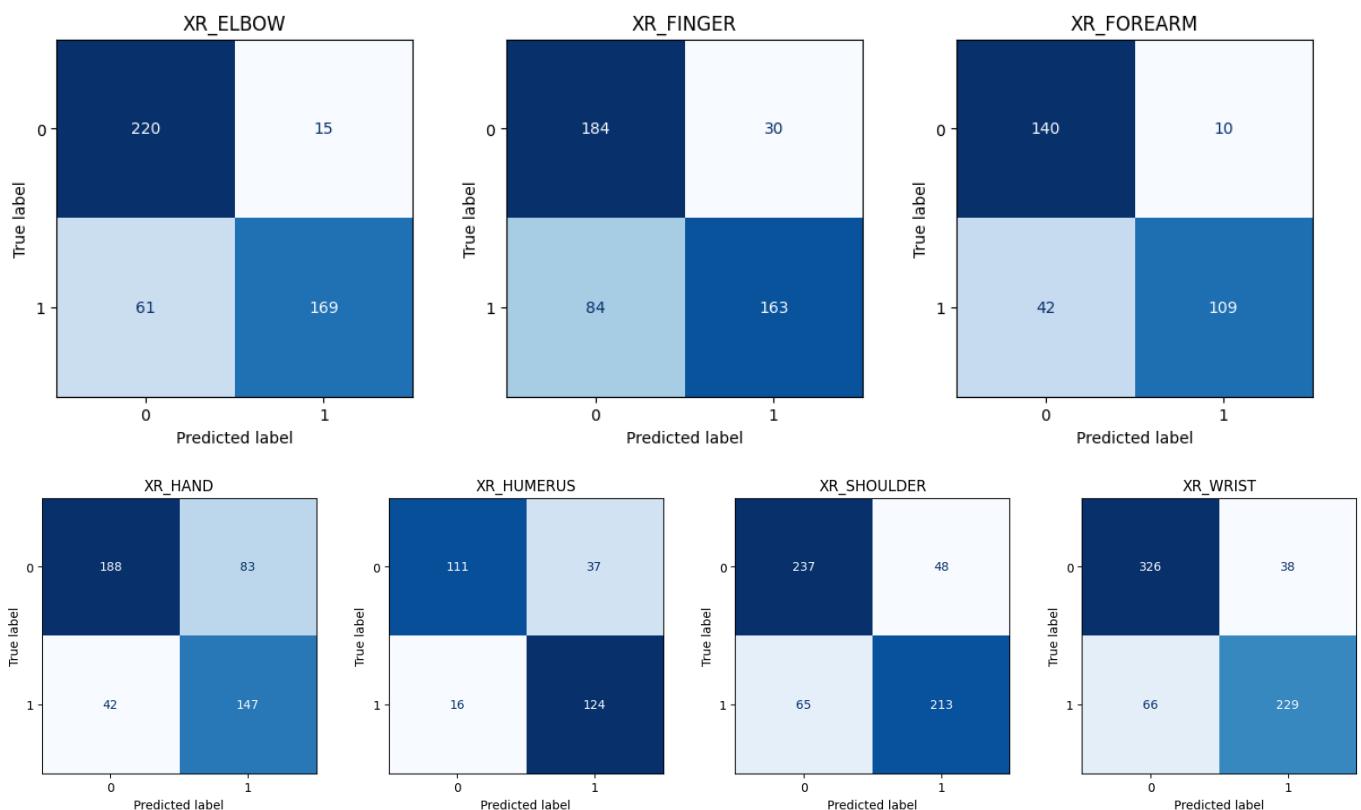


Рисунок 21 - Confusion Matrices

Для качественной интерпретации результатов помимо ROC-AUC использовались матрицы ошибок, рассчитанные при фиксированном пороге классификации $t=0.5$ (см. Рисунок 21). Матрица ошибок позволяет разделить ошибки на ложноположительные (FP: норма ошибочно распознана как патология) и ложноотрицательные (FN: пропуск патологии). Эти типы ошибок имеют разную практическую цену: FN критичны для задач скрининга и первичного отбора, тогда как FP увеличивают нагрузку на врача, снижая специфичность.

Анализ по анатомическим областям показывает, что baseline DenseNet демонстрирует разные профили ошибок: для одних областей модель ведет себя более консервативно (низкий FP, но повышенный FN), для других - более чувствительно (низкий FN при заметном FP). Это важно, поскольку единый порог не обязательно соответствует оптимальной рабочей точке для всех областей.

XR_ELBOW. Для локтя наблюдается низкое число FP (15) при сравнительно заметных FN (61). Такой профиль соответствует высокой специфичности и точности предсказаний «патология», однако часть патологических случаев остается не выявленной. Это указывает, что в данной области модель склонна выдавать положительный ответ только при высокой уверенности, и фиксированный порог $t=0.5$ может быть избыточно строгим для режима «не пропускать патологию». Практически это означает необходимость подбора порога под целевой режим (например, повышение чувствительности) и/или методов, уменьшающих число FN.

XR_HUMERUS. Для плечевой кости характерен противоположный паттерн: FN мало (16), но FP существенно больше (37). Модель демонстрирует высокую чувствительность, но чаще выдает ложные тревоги на нормальных исследованиях. Такой профиль соответствует режиму «чувствительного детектора», полезного для триажа/приоритизации (поднять подозрительные исследования выше), но потенциально увеличивающего число лишних направлений на перепроверку из-за FP.

XR_FOREARM. Для предплечья наблюдается консервативный режим: FP крайне мало (10), при этом FN остаются заметными (42). Модель уверенно подтверждает патологию, когда признаки выражены, но может не распознать слабо выраженные/пограничные случаи. Это типично, когда разделимость классов в целом высокая, но часть патологий имеет низкую контрастность или проявляется локально и теряется в глобальной агрегации признаков.

XR_WRIST. Запястье демонстрирует более сбалансированное соотношение FP (38) и FN (66) при достаточно большом числе верных классификаций в обоих классах. В сравнении с другими областями это ближе к рабочей точке при $t=0.5$: модель не только ранжирует случаи (что отражается в AUC), но и дает приемлемый компромисс между чувствительностью и специфичностью без сильного перекоса в одну сторону. Это делает область перспективной для дальнейшего улучшения (калибровка вероятностей, подбор порога, усиленные аугментации).

XR_SHOULDER. Для плечевого сустава FP (48) и FN (65) сопоставимы, что указывает на отсутствие ярко выраженного перекоса: модель ошибается в обе стороны. Подобная картина часто связана с высокой внутриклассовой вариативностью: различия проекций/укладки, наложения тканей, сложная анатомия области. В этом случае улучшения чаще достигаются за счет повышения устойчивости к вариативности (аугментации, регуляризация, более выразительные архитектуры, per-study агрегация).

XR_FINGER. Для пальцев основной источник ошибок - FN (84) при умеренном FP (30). Это означает, что модель чаще пропускает патологию, чем перестраховывается. Вероятная причина - более мелкий масштаб диагностически значимых признаков (тонкие линии перелома, слабый контраст), из-за чего при приведении к фиксированному размеру и стандартных преобразованиях модель может терять локальные детали. Данный профиль обосновывает использование методов, усиливающих обучение на локальных паттернах (например, стратегии внимания/erasing, более сильные аугментации и/или трансформерные компоненты).

XR_HAND. Для кисти критичной проблемой является высокий FP (83) при сравнительно неплохом FN (42). Это указывает, что модель склонна реагировать на неспецифические визуальные паттерны, коррелирующие с классом «патология» в обучении (например, особенности проекций, тени, артефакты), что приводит к ложным тревогам. Для этой области особенно актуальны методы, которые «переводят» внимание модели с фоновых коррелятов на более устойчивые признаки (например, MixUp/CutMix, CAM-erasing и анализ карт внимания).

Baseline DenseNet не демонстрирует единого типа ошибок на всех областях: в одних случаях модель более консервативна, в других - более чувствительна. Следовательно, улучшение качества следует рассматривать не только как рост AUC, но и как коррекцию профиля ошибок в целевом операционном режиме: для задач скрининга - снижение FN, для снижения нагрузки на врача - снижение FP. Это обосновывает дальнейшие подходы, использованные в работе.

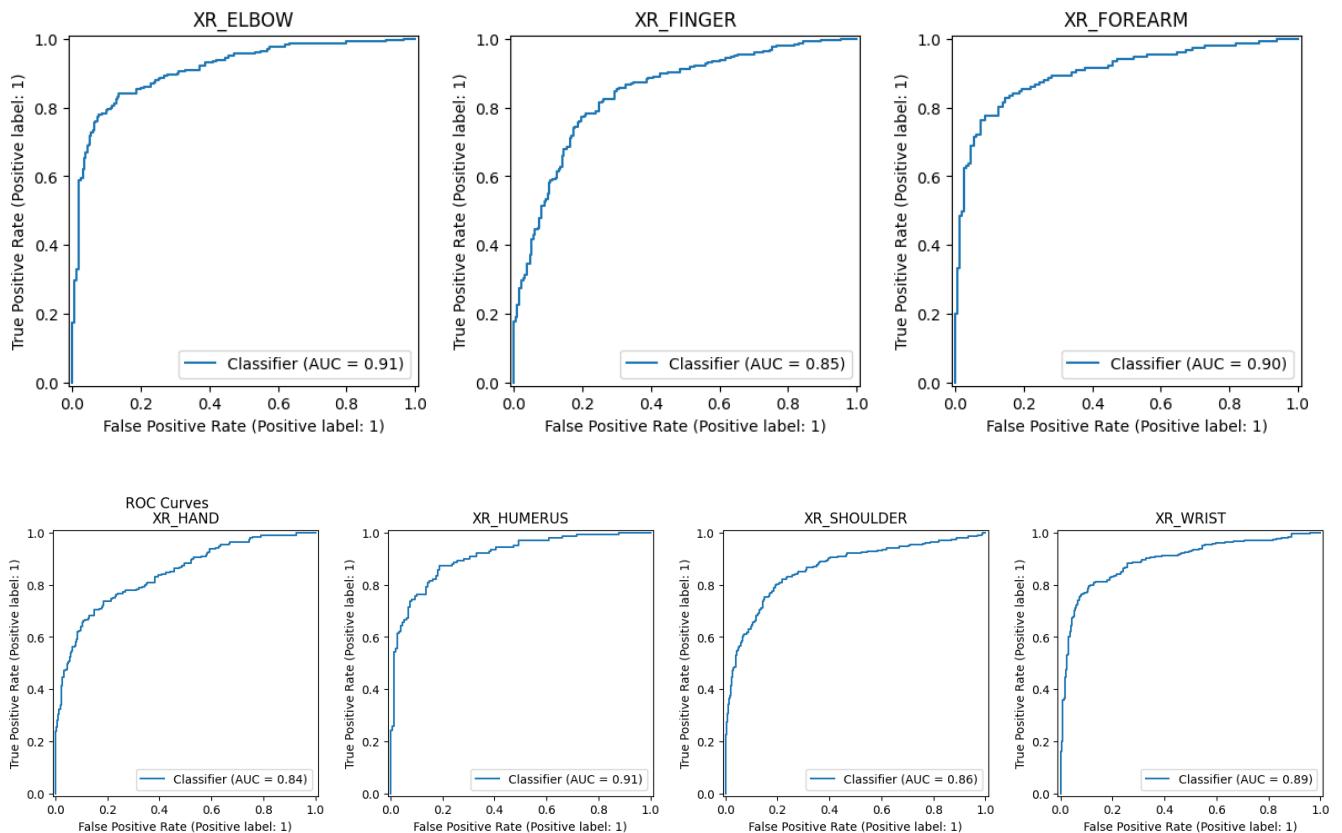


Рисунок 22 - ROC Curves

ROC-AUC характеризует способность модели ранжировать примеры: вероятность (score) для патологических исследований должна быть статистически выше, чем для нормальных. ROC-кривая строится по всем возможным значениям порога и показывает компромисс между чувствительностью (TPR) и долей ложных срабатываний (FPR). Следовательно, высокий ROC-AUC означает, что модель в целом хорошо различает классы, однако не гарантирует оптимального качества при конкретном пороге, используемом для принятия решения.

Полученные результаты (см. Рисунок 22) демонстрируют, что при сопоставимо высоких значениях ROC-AUC по ряду областей профиль ошибок существенно различается. Анализ подтверждает наличие информативного сигнала и способность модели различать классы, а матрицы ошибок показывают практическую пригодность выбранного порога. В прикладных медицинских сценариях порог целесообразно подбирать под требуемый режим работы (например, обеспечивая ограничение снизу на чувствительность $\text{Recall} \geq X$ для задач скрининга) и только после этого сравнивать модели по дополнительным метрикам.

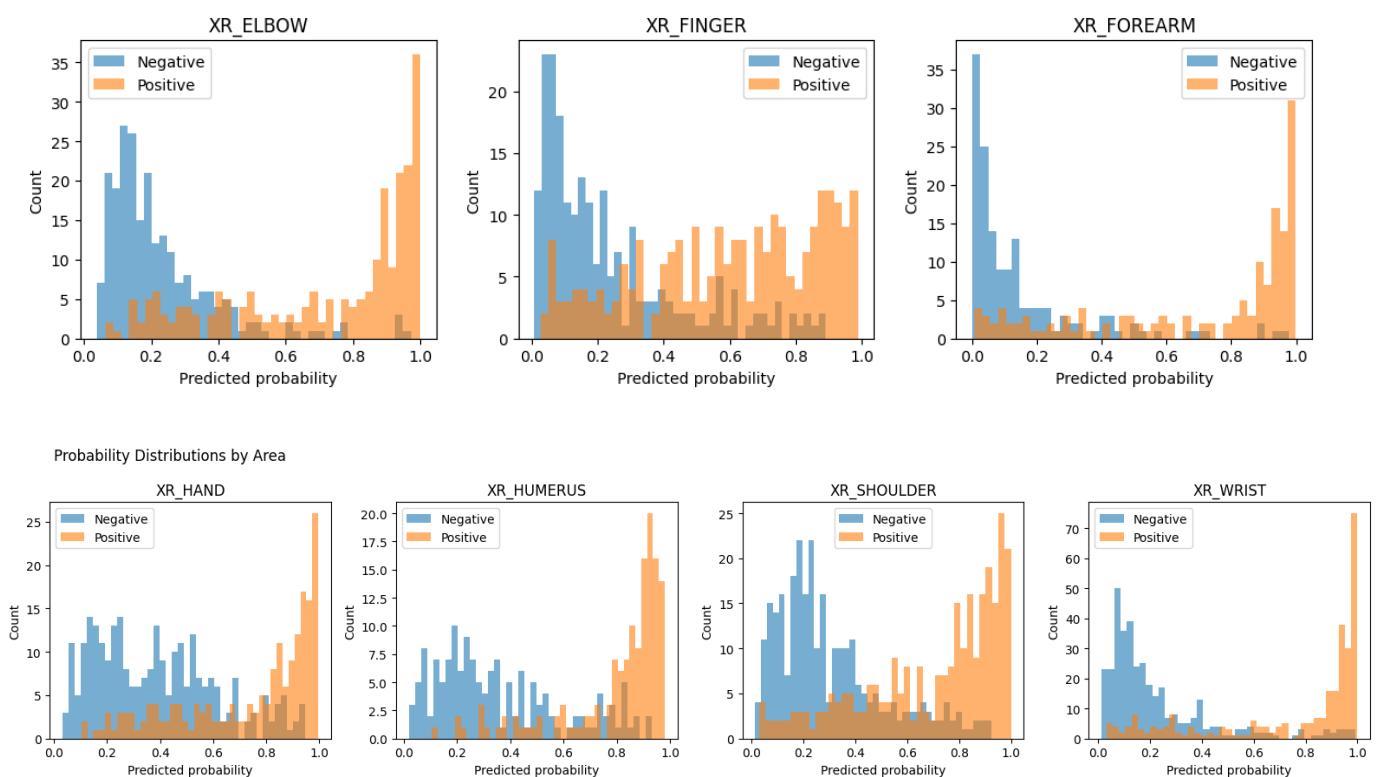


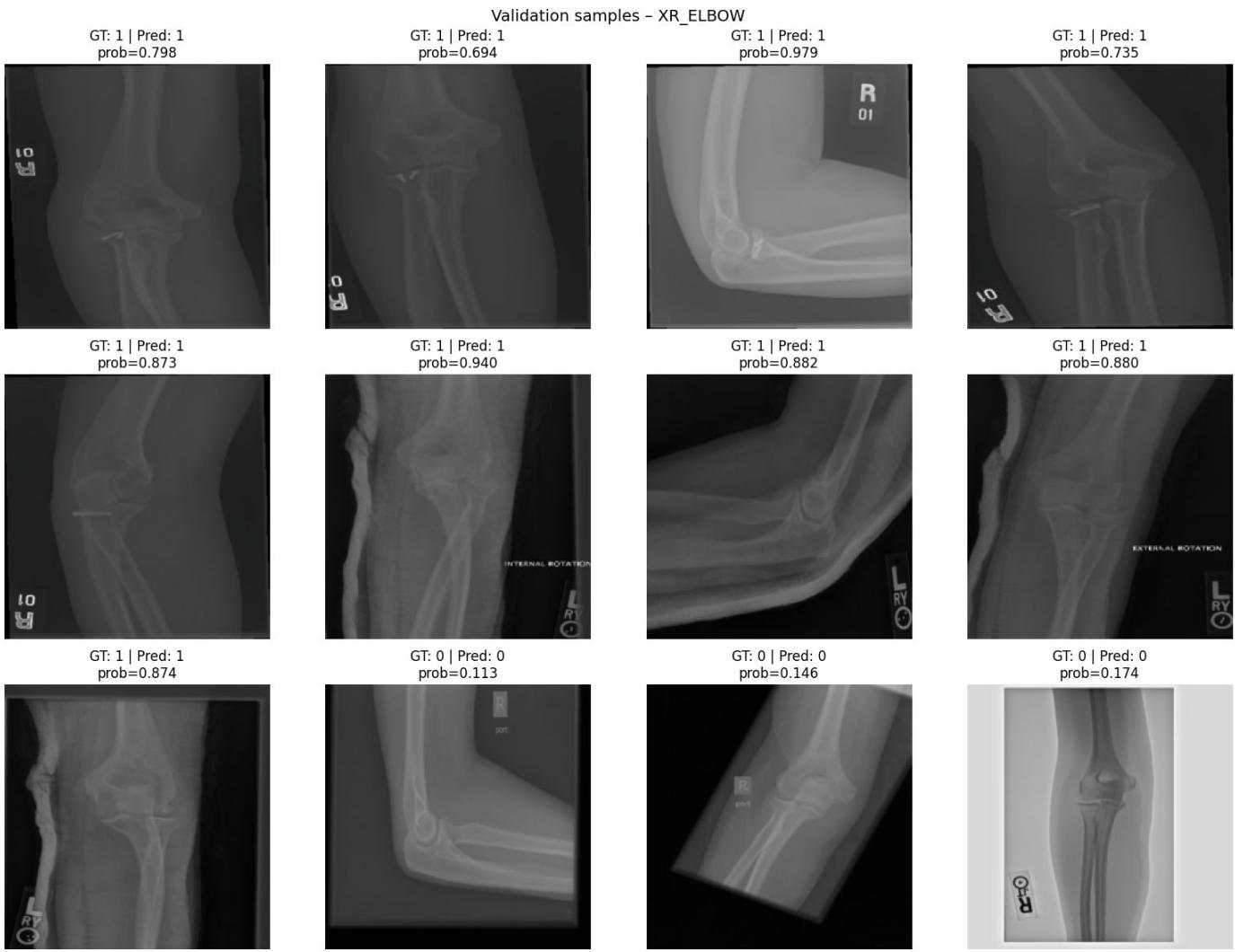
Рисунок 23 - Probability Distributions by Area

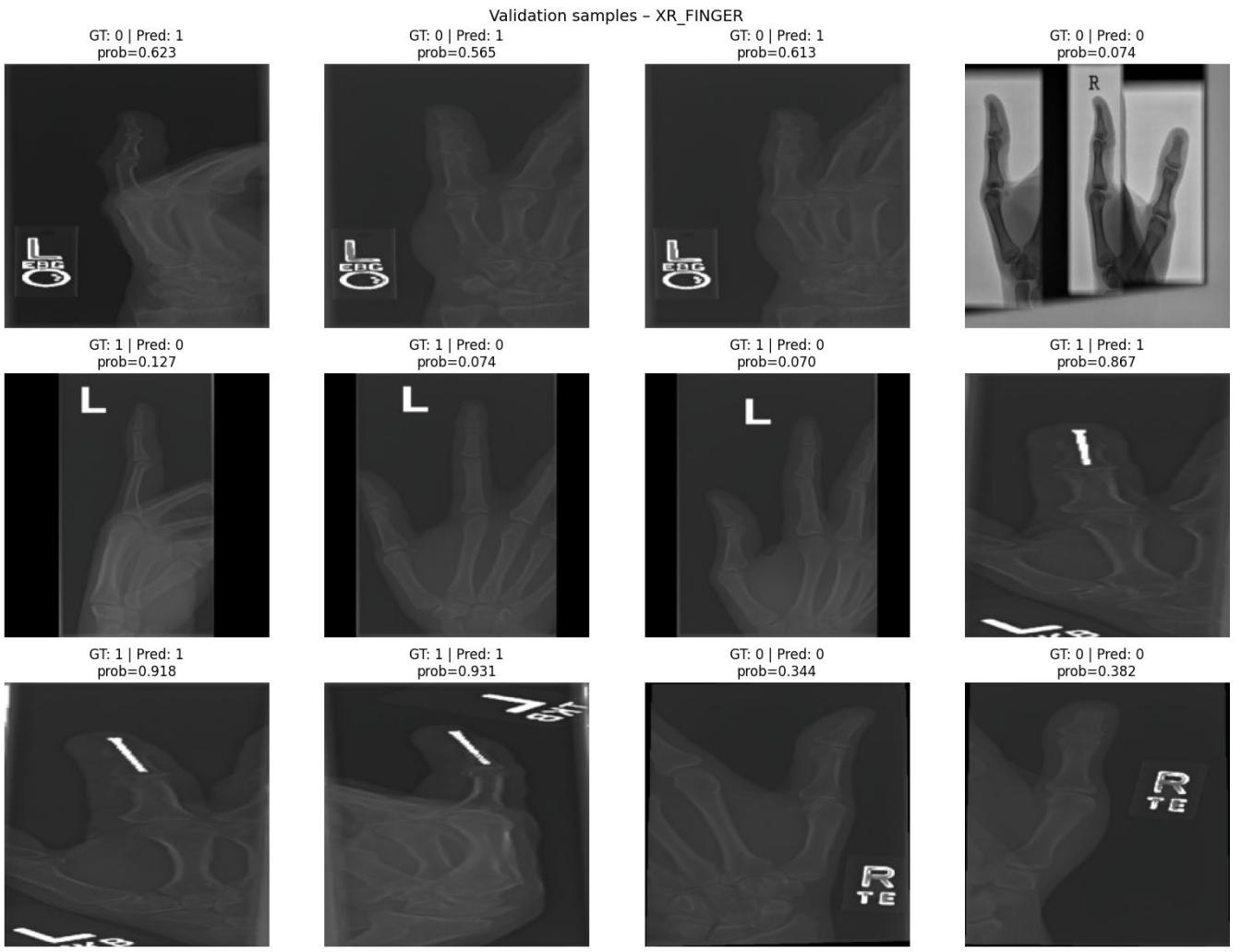
Анализ распределений предсказанных вероятностей (Probability Distributions) позволяет интерпретировать поведение модели beyond aggregate-метрик, поскольку напрямую показывает перекрытие скорингов классов и цену выбранного порога классификации. В случаях, когда модель формирует хорошо разделенные распределения для классов, ROC-AUC, как правило, оказывается высоким, а выбор операционной точки (например, $\text{threshold}=0.5$) дает приемлемый баланс между чувствительностью и специфичностью. Напротив, заметное перекрытие

распределений означает, что даже при хорошем ранжировании часть примеров попадает в зону неопределенности, где ошибки будут зависеть от выбора порога.

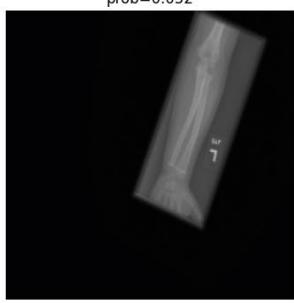
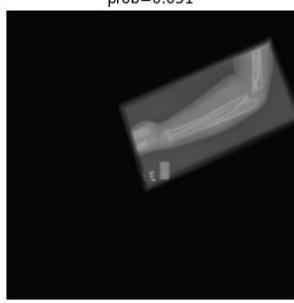
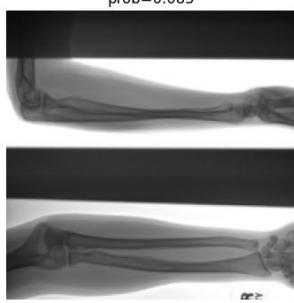
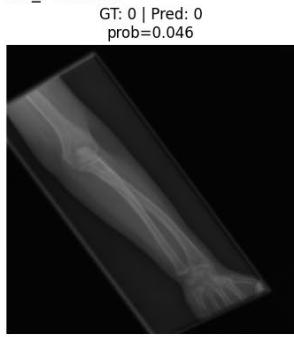
Для XR_WRIST, XR_ELBOW и XR_FOREARM наблюдается близкая к бимодальной структура распределений: отрицательный класс преимущественно сосредоточен в диапазоне низких вероятностей ($\approx 0\text{--}0.2$), а положительный - в области высоких вероятностей ($\approx 0.8\text{--}1.0$). Такое разделение свидетельствует о высокой уверенности модели и объясняет лучшие показатели качества для этих областей. Для XR_FINGER распределение положительного класса значительно расширено и частично смещено в средний диапазон ($\approx 0.3\text{--}0.6$), формируя существенное перекрытие с отрицательным классом. Это приводит к росту доли ложноотрицательных решений (FN) при фиксированном пороге 0.5, что согласуется с результатами, полученными на матрице ошибок. Для XR_HAND, напротив, характерен «правый хвост» у отрицательного класса: часть нормальных исследований получает средние и высокие значения predicted probability. Данный эффект указывает на слабую специфичность в области кисти и объясняет рост FP; вероятной причиной является высокая внутриклассовая вариативность и наличие визуальных факторов, коррелирующих с меткой патологии (наложения, проекции, артефакты), но не являющихся самой патологией.

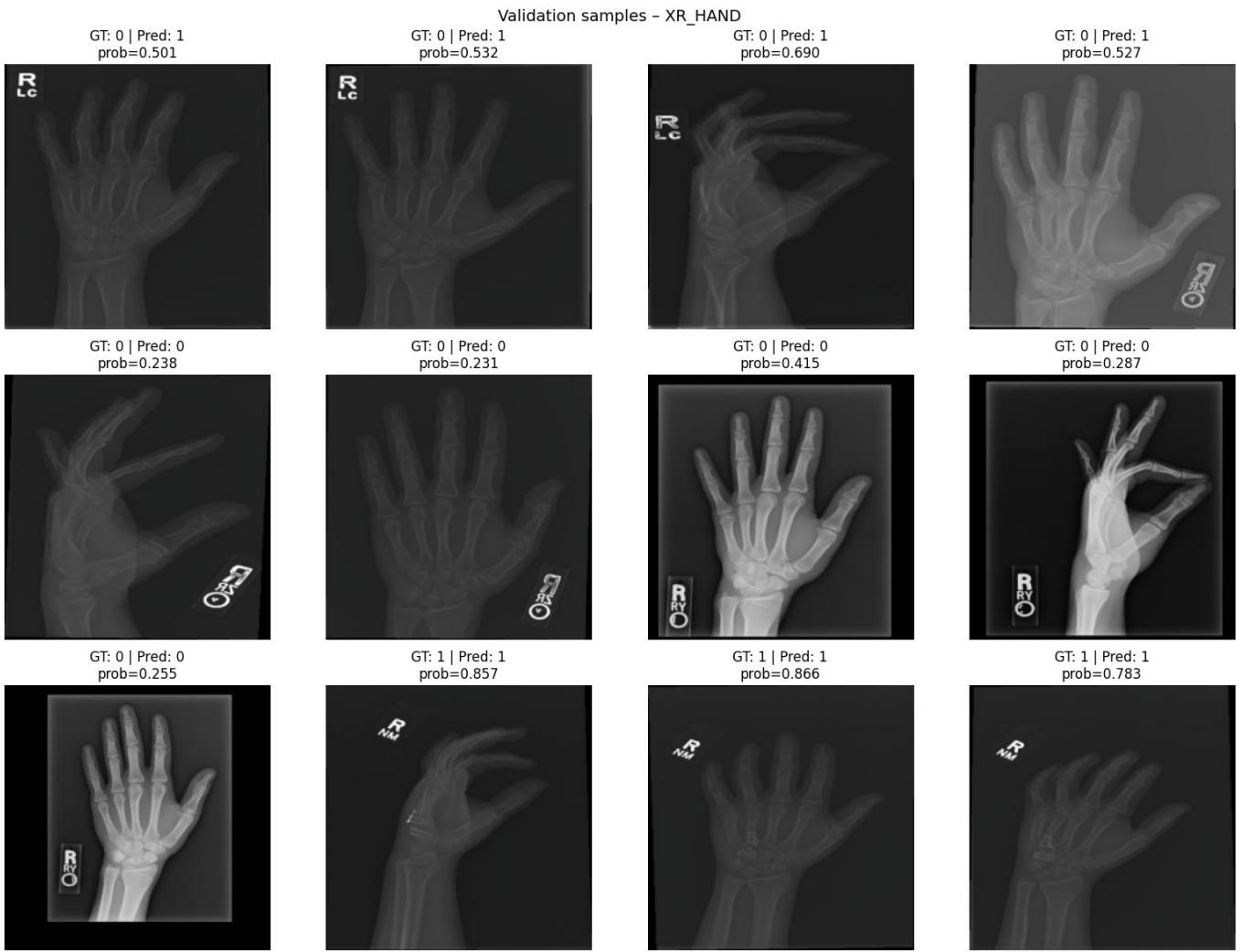
Выход sigmoid не гарантирует калиброванность вероятностей: значения модели отражают скоринговую уверенность, однако могут систематически переоценивать или недооценивать истинный риск. Поэтому сильно правдоподобные вероятности не обязательно означают корректные вероятностные оценки. Для медицинских сценариев, где важна интерпретация уверенности, целесообразно дополнительно рассматривать калибровочные кривые (reliability diagrams), метрики калибровки (например, ECE) и, при необходимости, применять посткалибровку (temperature scaling) на валидационном наборе.

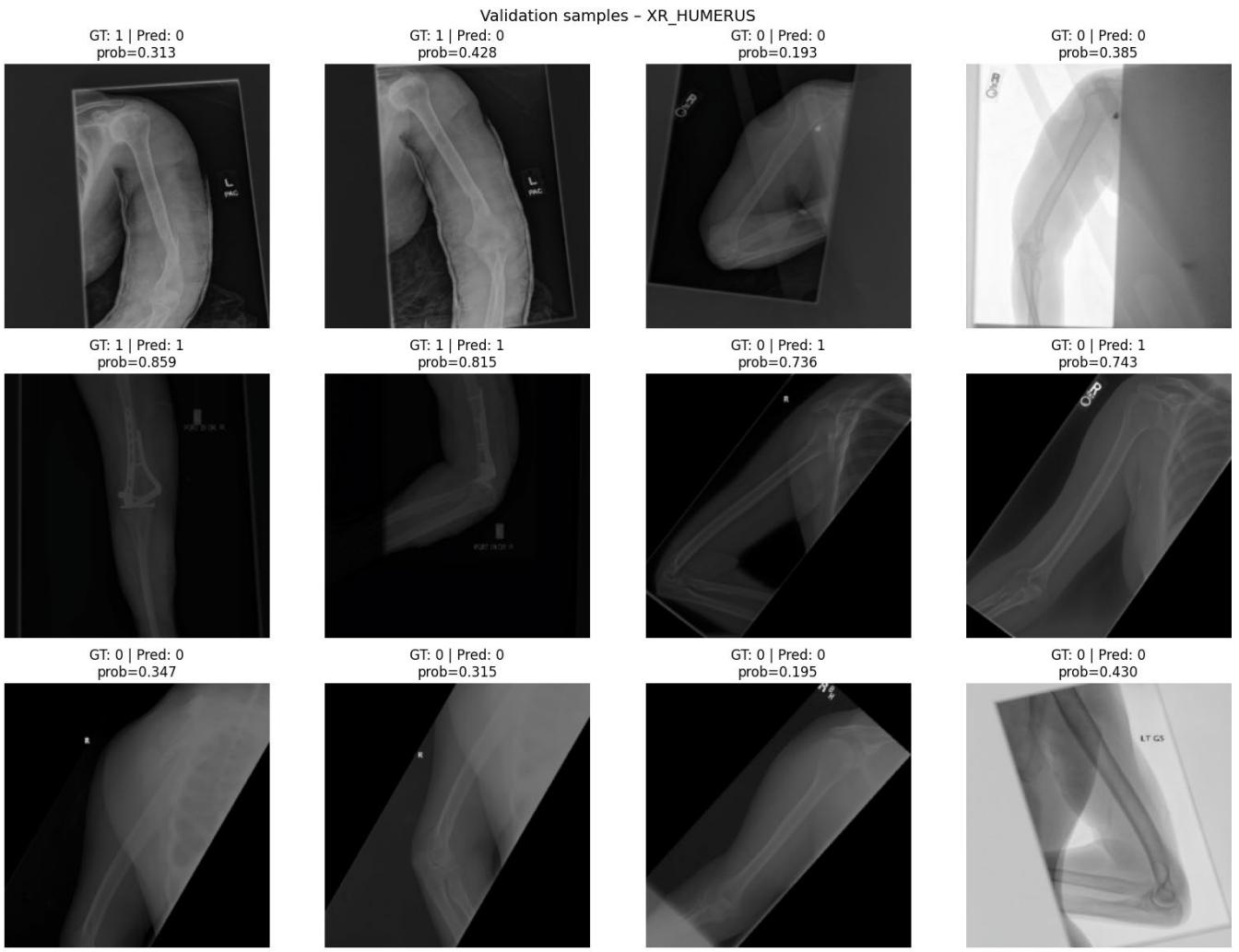




Validation samples - XR_FOREARM







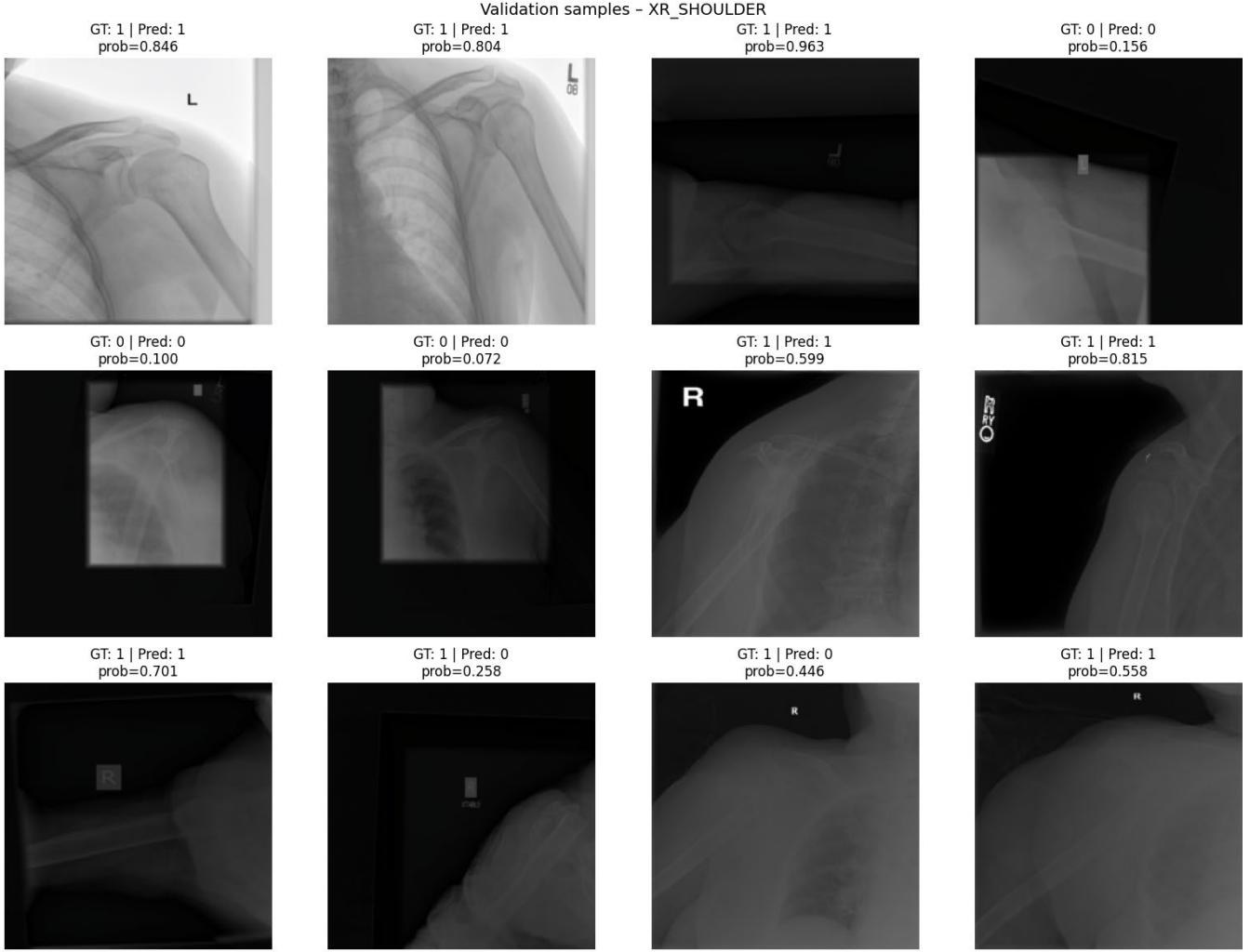




Рисунок 24 - "XR_ELBOW", "XR_FINGER", "XR_FOREARM", "XR_HAND", "XR_HUMERUS", "XR_SHOULDER", "XR_WRIST"

XR_ELBOW

На примерах видно: для abnormal модель часто дает уверенные $p \approx 0.7-0.98$, для normal - низкие $p \approx 0.11-0.17$. Это типичный профиль: precision высокий, потому что модель редко восходит в positive без уверенности. Локтевые переломы на некоторых проекциях могут быть тонкими, плюс встречаются сложные случаи (перекрытия, частичная визуализация сустава, границы коллимации), где сеть не набирает $p > 0.5$.

Вывод: в локте baseline работает как детектор явных abnormal, но теряет subtle cases.

XR_FOREARM

На примерах отрицательные случаи имеют низкие вероятности ($\approx 0.01-0.08$) - это признак хорошей специфичности. Есть и подозрительные нормальные/пограничные кадры (например $p \approx 0.38$ при GT0) - обычно это связано с нестандартной укладкой, экспозицией, рамками/масками, наличием фиксирующих конструкций вне зоны интереса.

Вывод: сеть хорошо отсекает норму, но часть патологии в предплечье может быть тонкой $\rightarrow FN$ остаются.

XR_WRIST

В примерах почти все abnormal имеют $p \approx 0.93-0.99$, один случай GT1 \rightarrow Pred0 с $p \approx 0.28$ - классический subtle fracture / сложная проекция / слабый контраст. При этом видно, что изображения с металлоконструкциями дают экстремально высокие p - это намекает на возможную эвристику hardware \Rightarrow abnormal. Для MURA это часто корректно (abnormal включает постоперационные состояния), но клинически это может быть не равно наличию перелома сейчас.

Вывод: запястье - сильная область, но baseline может быть слишком уверенным на конфаундерах типа фиксаторов.

XR_FINGER

В примерах есть: несколько FP с умеренной уверенностью (GT0 Pred1, $p \approx 0.56-0.62$) - так бывает из-за наложения фаланг, артефактов, маркеров, кривой экспозиции; несколько FN с очень низкой уверенностью (GT1 Pred0, $p \approx 0.07-0.13$) - это сильный показатель, что модель вообще не увидела патологию на этих кадрах. Частая причина именно для пальцев: объект маленький, признаки тонкие, а при ресайзе до 224×224 микродетали просто исчезают.

Вывод: для пальцев baseline упирается в разрешение/локализацию и вариативность позы → это идеальная цель для подхода 2-3 и/или увеличения входного размера/patch-ориентированных моделей.

XR_HAND

В первой строке подряд идут ложноположительные с $p \approx 0.50-0.69$. Модель реагирует на вещи, которые коррелируют с abnormal в трейне, но не являются переломом:

- сильное наложение костей/пальцев,
- движение/смаз,
- необычная укладка кисти,
- контрастные края, рамки, маркеры.

Вывод: baseline в HAND склонен к гипернастороженности; в улучшениях логично давить конфаундеры (MixUp/CutMix), а также заставлять сеть искать альтернативные признаки (CAM-erasing).

XR_HUMERUS

На примерах видны FN с $p \approx 0.31-0.43$, и FP с $p \approx 0.74$. Плечевая кость часто снимается с большим количеством окружающих тканей/сустава/частичной визуализации → появляются нестабильные паттерны и больше confounders.

Вывод: HUMERUS baseline ловит большинство abnormal (recall высокий), но точность падает из-за ложных тревог.

XR_SHOULDER

Плечо - одна из самых «грязных» зон по визуальному разнообразию: грудная клетка, ключица, лопатка, разные проекции, низкий контраст. В примерах видно, что

модель иногда предсказывает positive с умеренной уверенностью ($p \approx 0.56$ - 0.60), а иногда промахивается (пример FN около $p \approx 0.45$). Это указывает на то, что граница классов проходит как раз в серой зоне вероятностей.

Вывод: для плеча сильнее всего поможет либо улучшение данных/аугментаций, либо более внимательная архитектура/агрегация по study.

Результаты модели с НРО и CAM

area	best_auc_study	lr	weight_decay	mixup_alpha	cutmix_alpha	aug_p	erase_q	alpha_cam	erase_every
XR_HUMERUS	0.933275	0.000086	2.733764e-06	0.244134	0.298984	0.666289	0.824012	0.671039	6
XR_WRIST	0.909352	0.000063	1.187844e-04	0.120894	0.151029	0.512462	0.700714	0.623986	10
XR_ELBOW	0.908762	0.000036	6.351221e-04	0.439196	0.897988	0.378009	0.731199	0.323233	12
XR_FOREARM	0.898777	0.000200	6.502726e-07	0.206339	0.610838	0.760945	0.801239	0.395039	9
XR_FINGER	0.860922	0.000133	2.586200e-06	0.564747	1.163579	0.306570	0.772926	0.559164	10
XR_SHOULDER	0.849867	0.000107	1.682893e-04	0.116072	1.101334	0.657585	0.831634	0.445500	9
XR_HAND	0.826283	0.000059	1.224109e-07	0.493478	0.017368	0.626502	0.837907	0.354872	11

Рисунок 25 - Optuna summary

В рамках подхода 2 гиперпараметры подбирались автоматически с помощью Optuna отдельно для каждой анатомической области. Итогом оптимизации является таблица лучших конфигураций, где best_auc_study отражает максимальное значение AUC на уровне исследования (study-level), достигнутое для данной области, а остальные столбцы содержат выбранные Optuna значения гиперпараметров.

Полученные значения подтверждают, что оптимальные режимы обучения существенно различаются между областями, то есть универсальные гиперпараметры были бы заведомо компромиссными. Например, для XR_HUMERUS Optuna выбирает относительно умеренную частоту CAM-erasing (erase_every=6) при достаточно заметном весе erasing-компоненты ($\text{alpha_cam} \approx 0.67$) и $\text{erase_q} \approx 0.82$, что означает регулярное подавление зон внимания и стимулирование модели искать дополнительные признаки помимо наиболее очевидных. Для XR_ELBOW наблюдается более редкий erasing (erase_every=12) и меньший вес $\text{alpha_cam} \approx 0.32$, при этом MixUp/CutMix используются интенсивнее (высокие mixup_alpha и особенно cutmix_alpha), что указывает на большую роль регуляризации по данным, чем агрессивной регуляризации внимания. Для XR_HAND, являющейся наиболее сложной областью, Optuna, напротив, выбирает

практически отключённый CutMix ($\text{cutmix_alpha} \approx 0.02$) при достаточно высоком MixUp ($\text{mixup_alpha} \approx 0.49$) и умеренной частоте erasing ($\text{erase_every}=11$): это согласуется с тем, что крупные прямоугольные вставки CutMix могут чаще портить локальные структуры мелких костей кисти и ухудшать обучающий сигнал. В целом, разброс параметров по областям демонстрирует, что различия в визуальной сложности, масштабе анатомических структур и типичных артефактах требуют разных балансов между оптимизацией, аугментациями и CAM-erasing.

area	val_size_img	AUC_img	ACC_img	Precision_img	Recall_img	F1_img	AUC_study	ACC_study	Precision_study	Recall_study	F1_study	AUC_stored_best_study
XR_HUMERUS	288	0.9143	0.8299	0.7862	0.8929	0.8361	0.9333	0.8741	0.8289	0.9403	0.8811	0.9333
XR_WRIST	659	0.8864	0.8270	0.9209	0.6712	0.7765	0.9094	0.8523	0.9429	0.6804	0.7904	0.9094
XR_ELBOW	465	0.8988	0.7935	0.9467	0.6174	0.7474	0.9088	0.8038	0.9070	0.5909	0.7156	0.9088
XR_FOREARM	301	0.8638	0.7841	0.8909	0.6490	0.7510	0.8988	0.7970	0.8936	0.6562	0.7568	0.8988
XR_FINGER	461	0.8459	0.7766	0.8273	0.7368	0.7794	0.8609	0.7886	0.8194	0.7108	0.7613	0.8609
XR_SHOULDER	563	0.8389	0.7851	0.8428	0.6942	0.7613	0.8499	0.7835	0.8354	0.6947	0.7586	0.8499
XR_HAND	460	0.7967	0.7261	0.7368	0.5185	0.6087	0.8263	0.7725	0.8500	0.5152	0.6415	0.8263

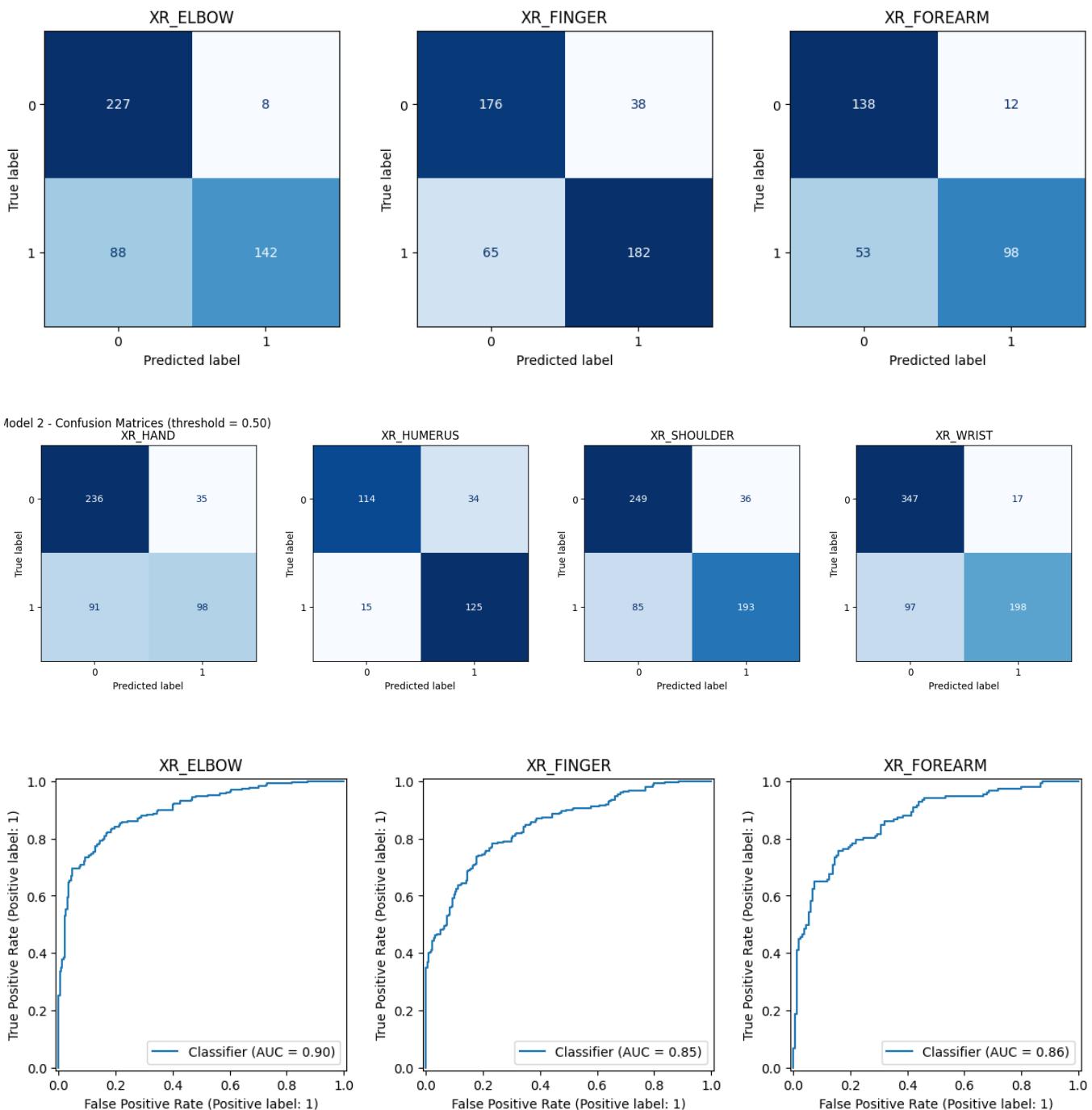
Рисунок 26 - MODEL 2 - METRICS (image + study level)

В таблице представлены метрики качества для каждой анатомической области на валидации. Показаны две группы показателей: image- и study-level. Ключевое наблюдение по таблице - во всех областях AUC_study выше AUC_img. Это типичный эффект для MURA-пайплайна: даже если на некоторых проекциях модель не уверена, агрегирование по исследованию повышает устойчивость ранжирования и лучше соответствует клиническому сценарию, где решение принимается по совокупности проекций. Дополнительно столбец AUC_stored_best_study подтверждает, что сохраненный лучший чекпоинт соответствует достигнутому значению AUC_study.

Если смотреть на абсолютные результаты, лидером является XR_HUMERUS: AUC_study ≈ 0.933 при высокой полноте (Recall_study ≈ 0.94) и хорошем F1_study ≈ 0.88 . Такая комбинация означает, что модель не только хорошо ранжирует случаи (высокий AUC), но и при фиксированном пороге 0.5 демонстрирует чувствительное поведение (мало пропусков abnormal). Это согласуется с тем, что переломы плечевой кости визуально выражены сильнее и имеют более стабильный паттерн на снимках, поэтому регуляризация (MixUp/CutMix) и CAM-erasing здесь дают максимум пользы без сильного падения чувствительности. Группа XR_WRIST и XR_ELBOW показывает другую характерную картину: очень высокая точность

($\text{Precision}_{\text{study}} \approx 0.94$ и ≈ 0.91) при заметно более низкой полноте ($\text{Recall}_{\text{study}} \approx 0.68$ и ≈ 0.59). То есть модель в этих областях склонна быть консервативной: когда она говорит *abnormal*, это обычно действительно *abnormal* (низкий FP), но часть патологий она пропускает (высокий FN). Важно, что при этом $\text{AUC}_{\text{study}}$ остается высоким (≈ 0.909 и ≈ 0.909), а значит проблема не в способности разделять классы, а в выборе порога 0.5: распределения вероятностей и баланс ошибок здесь устроены так, что оптимальный порог для максимизации чувствительности/ F1 может отличаться от 0.5. Модель обладает высоким дискриминативным потенциалом (AUC), однако рабочая точка ($\text{threshold}=0.5$) приводит к смещению в сторону специфичности; для клинического применения целесообразна настройка порога под требуемый компромисс ошибок. XR_FOREARM выглядит более сбалансированной: $\text{Recall}_{\text{study}} \approx 0.66$ и $\text{Precision}_{\text{study}} \approx 0.89$ при $\text{AUC}_{\text{study}} \approx 0.899$. Здесь модель сохраняет высокое качество ранжирования и более ровный баланс ошибок, что отражается в стабильном $\text{F1}_{\text{study}} \approx 0.757$. По смыслу это устойчивая область: не максимальная по AUC, но без сильного перекоса в FP или FN. XR_FINGER и XR_SHOULDER - средний сегмент ($\text{AUC}_{\text{study}} \approx 0.861$ и ≈ 0.850). Здесь особенно заметно, что даже при неплохих Accuracy/F1 модель остается ограничена сложностью визуального сигнала: для пальцев и плеча типичны тонкие линии перелома, вариативность укладок и наложения тканей/структур, из-за чего граница между *normal* и *abnormal* менее контрастна. Для этих областей улучшения от HPO и регуляризации есть, но потолок вышеупомянутыми факторами достигается быстрее, чем для humerus/wrist/elbow. Наиболее проблемная область - XR_HAND: $\text{AUC}_{\text{study}} \approx 0.826$ и низкая полнота ($\text{Recall}_{\text{study}} \approx 0.52$) при высокой точности ($\text{Precision}_{\text{study}} \approx 0.85$). Это указывает на тот же консервативный режим, но уже на фоне более слабой разделимости классов. Практически это означает, что модель часто не набирает уверенность выше порога на *abnormal*-случаях (большая доля FN), что типично для кисти: множество мелких костей, наложения, высокая вариативность проекций и частые артефакты. Именно эта область логично мотивирует переход к более глобально-контекстным моделям или к стратегиям повышения разрешения обработки.

Отдельно стоит подчеркнуть методологический момент: AUC оценивает качество ранжирования без фиксированного порога, тогда как Precision/Recall/F1/Accuracy в таблице зависят от выбранного threshold=0.5. Поэтому корректная интерпретация такая: AUC показывает, насколько модель в принципе отделяет normal от abnormal, а метрики при пороге показывают выбранный компромисс FP/FN; при необходимости рабочая точка может быть сдвинута.



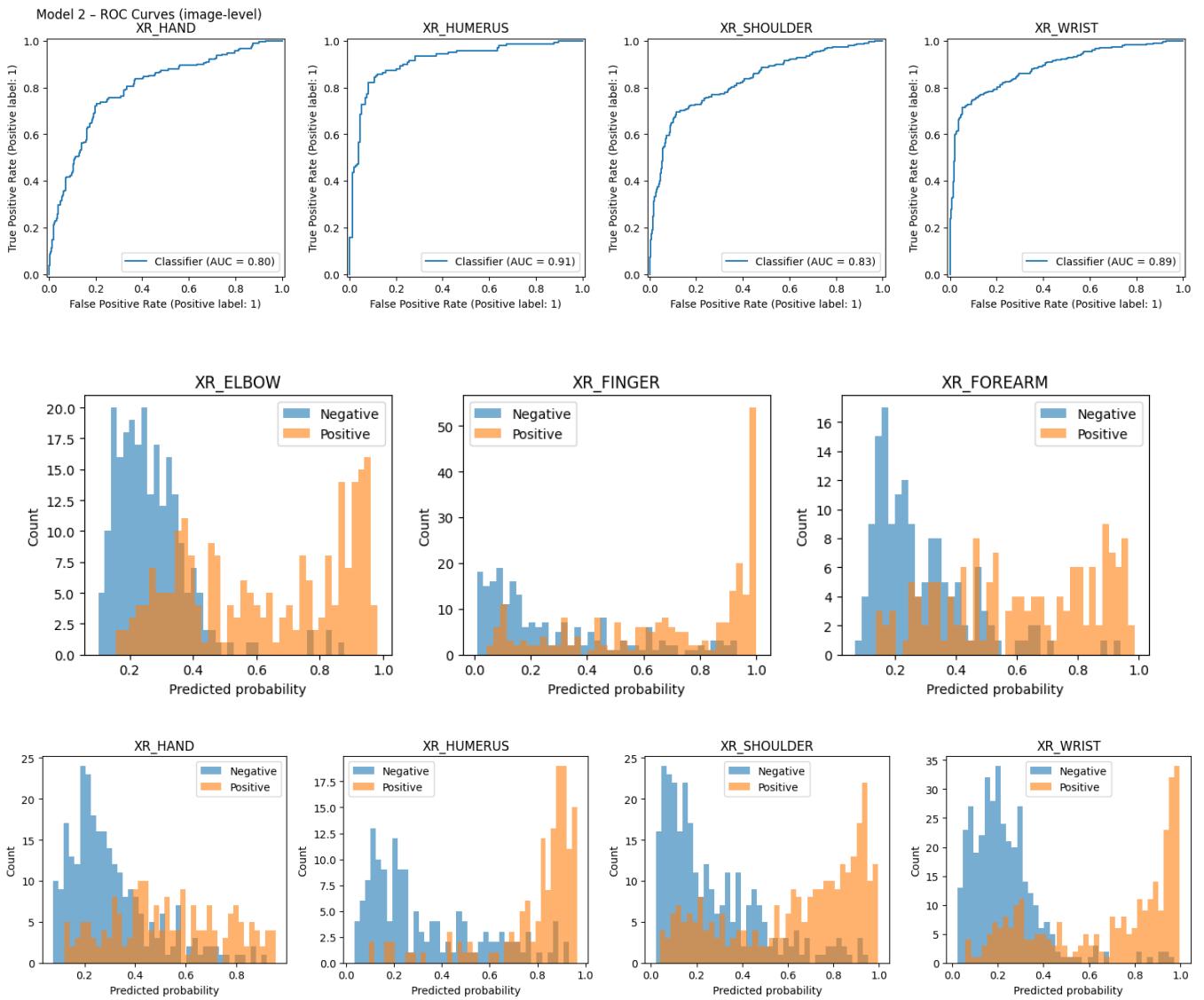


Рисунок 27 - Model 2 - Confusion Matrices, ROC Curves, Probability Distributions by Area

Большинство диагностических графиков удобно и информативно строить на image-level, потому что именно на этом уровне видно, как модель ведет себя на отдельных кадрах: где она сомневается, где выдает пограничные вероятности, и какие типичные ошибки возникают при пороге 0.5. Эти графики нужны как инструмент анализа качества скоринга и источников ошибок.

Важно, что AUC_img и AUC_study могут различаться. Например, аугментации MixUp/CutMix часто делают предсказания на отдельных кадрах менее резкими”(меньше экстремальных 0/1), из-за чего image-level AUC может расти незначительно или даже снижаться. Однако при агрегации на уровне исследования даже один информативный кадр способен вытянуть весь study в правильный класс,

и AUC_study становится выше. Поэтому, image-level графики используются для интерпретации поведения модели, а окончательные выводы о качестве делаются по study-level метрикам, соответствующим постановке задачи.

На рисунках представлены три типа диагностических графиков для анализа качества модели: матрицы ошибок при фиксированном пороге 0.5, ROC-кривые и распределения предсказанных вероятностей по классам. Для XR_ELBOW и XR_WRIST наблюдается консервативный режим классификации, что соответствует высокой специфичности и сниженной чувствительности при пороге. Для XR_HUMERUS, напротив, модель демонстрирует более скрининговый профиль, что соответствует смещению в сторону чувствительности. В случае XR_HAND одновременно выражены как FP, так и FN, что указывает на существенное перекрытие классов и повышенную сложность данных для данной области. ROC-кривые дополняют этот анализ, демонстрируя качество ранжирования независимо от выбранного порога: высокие значения AUC для XR_HUMERUS/XR_ELBOW/XR_WRIST свидетельствуют о хорошей разделимости на уровне скоринга, тогда как более низкий AUC для XR_HAND отражает ограничение модели в построении устойчивого ранжирования. Наконец, гистограммы вероятностей объясняют происхождение ошибок: наличие positive-примеров в промежуточном диапазоне ($\approx 0.3\text{--}0.6$) приводит к росту FN при пороге 0.5 (XR_WRIST/XR_ELBOW), тогда как хвост распределения negative вправо способствует FP (XR_HUMERUS). Для XR_HAND наблюдается максимальное перекрытие распределений классов, что согласуется с наихудшим профилем ошибок.

Validation samples - XR_ELBOW (Model 2)

GT: 1 | Pred: 1
prob=0.732



GT: 1 | Pred: 1
prob=0.764



GT: 1 | Pred: 1
prob=0.556



GT: 1 | Pred: 0
prob=0.343



GT: 1 | Pred: 1
prob=0.680



GT: 1 | Pred: 1
prob=0.609



GT: 1 | Pred: 1
prob=0.513



GT: 1 | Pred: 0
prob=0.378



GT: 1 | Pred: 0
prob=0.482



GT: 1 | Pred: 1
prob=0.609



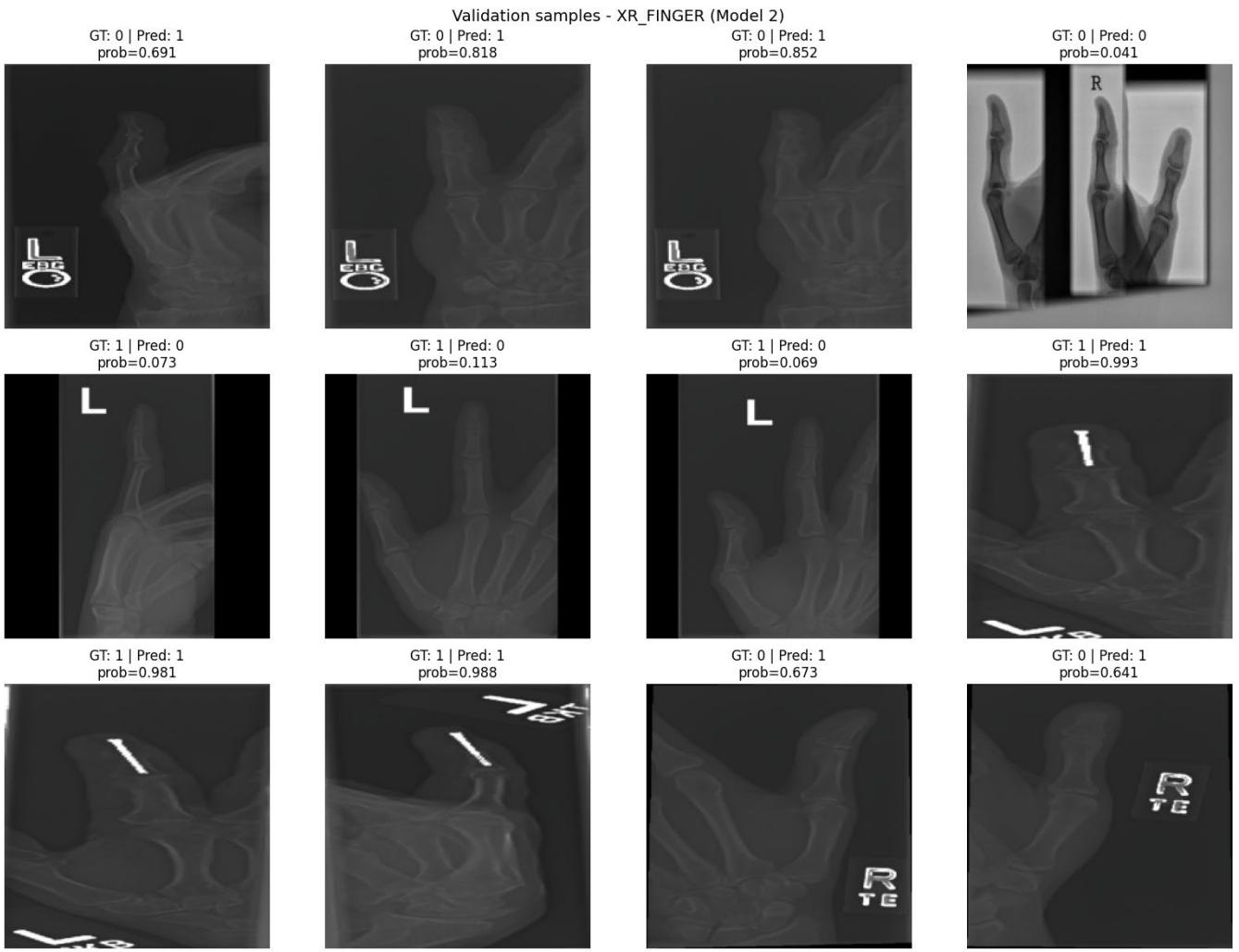
GT: 0 | Pred: 0
prob=0.264



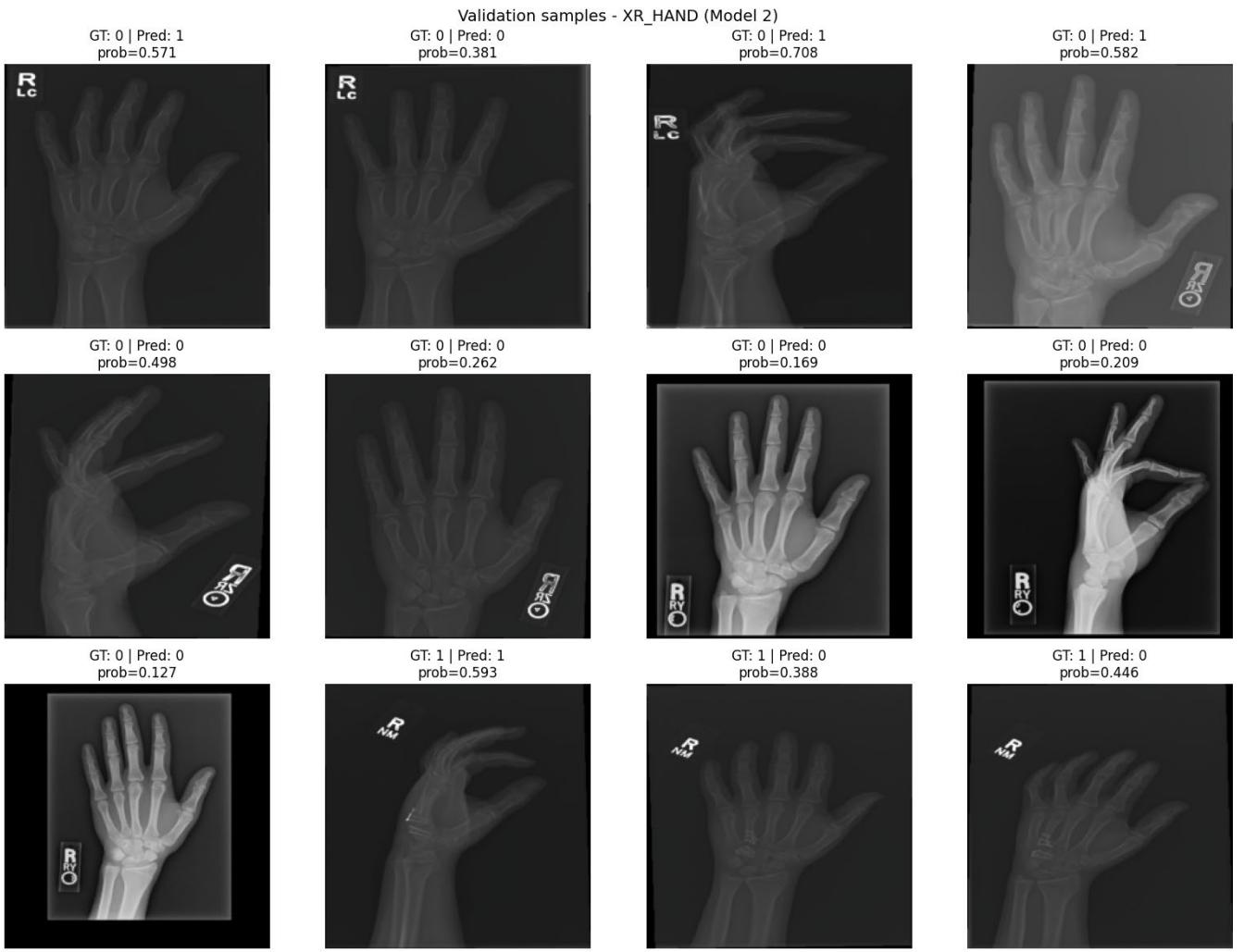
GT: 0 | Pred: 0
prob=0.144

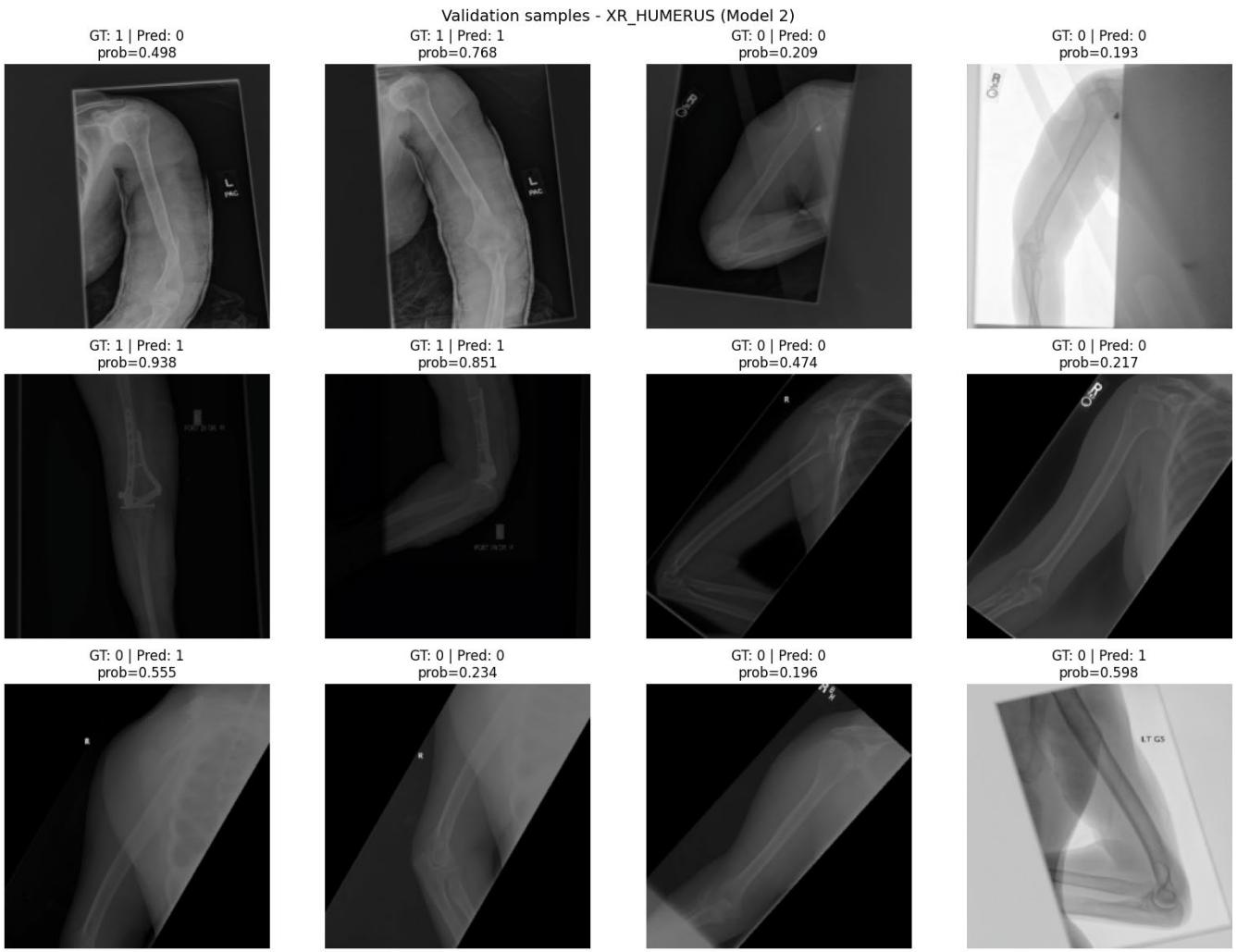


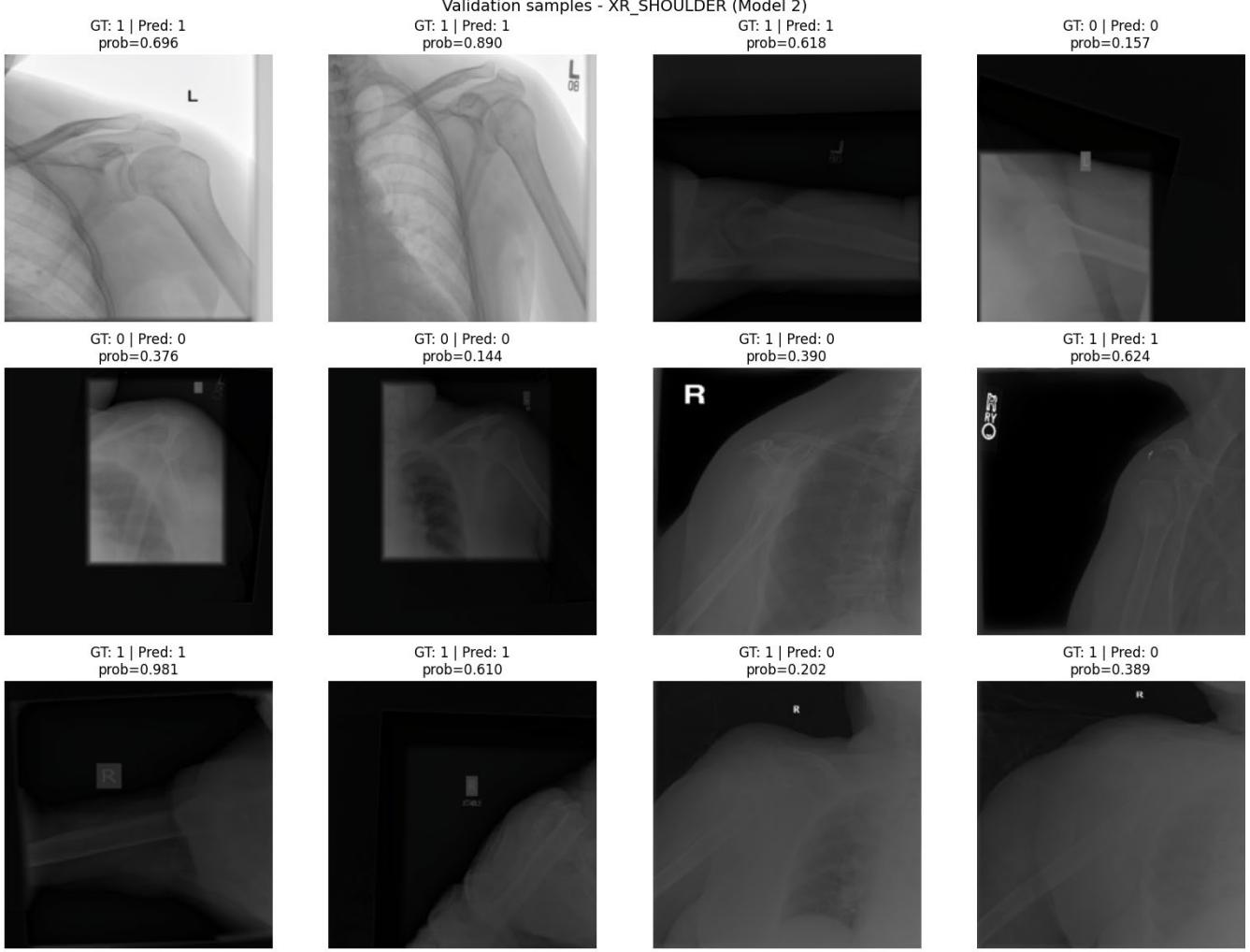
GT: 0 | Pred: 0
prob=0.390











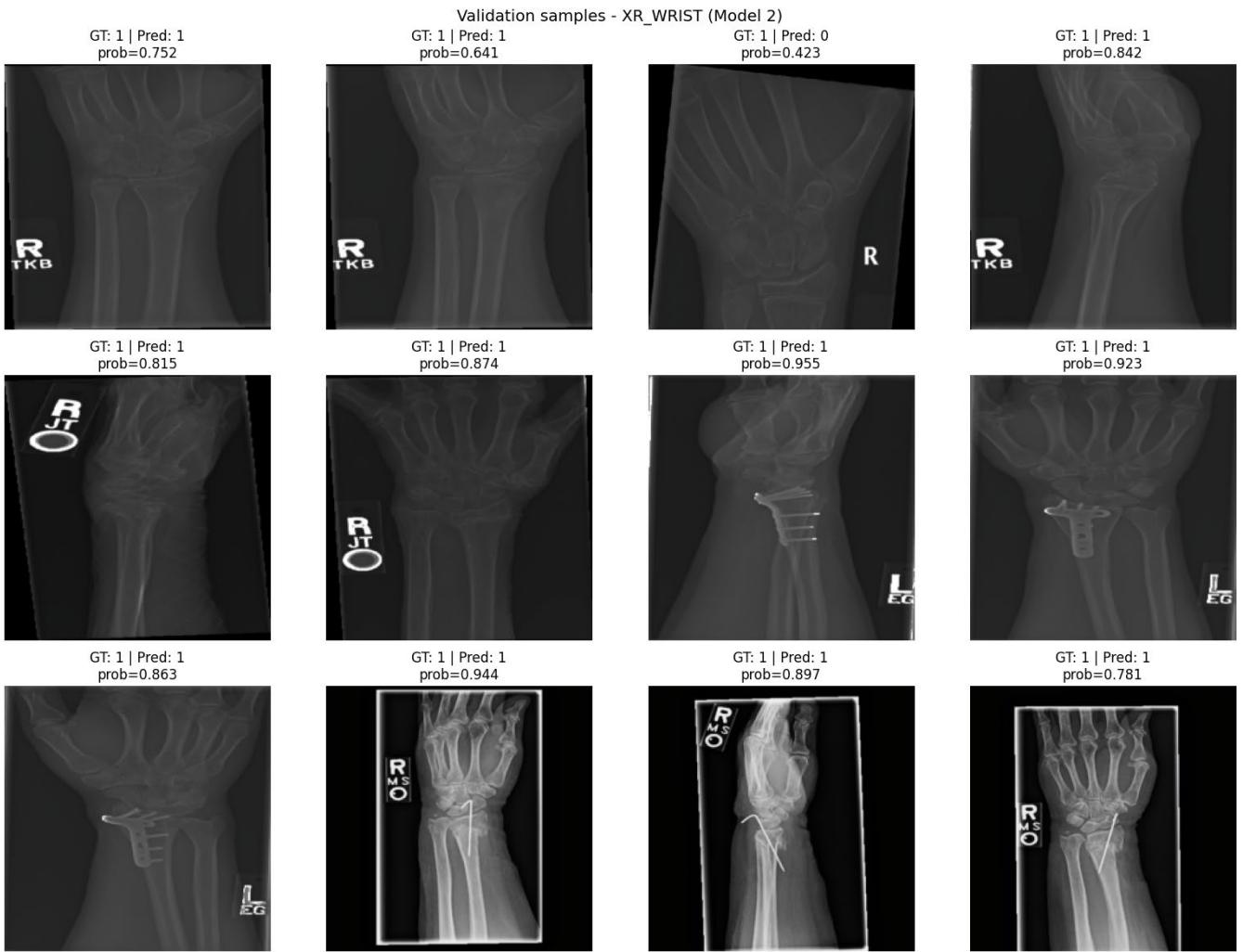


Рисунок 28 – 2 "XR_ELBOW", "XR_FINGER", "XR_FOREARM", "XR_HAND", "XR_HUMERUS", "XR_SHOULDER", "XR_WRIST"

Приведены примеры предсказаний модели (Model 2) на валидации для каждой анатомической области (см. Рисунок 28). Для каждого изображения показаны истинная метка (GT), бинарное решение при пороге 0.5 (Pred) и выходная вероятность после сигмоиды (prob). Данный qualitative-анализ дополняет количественные метрики, позволяя интерпретировать типичные сценарии ошибок (FP/FN) и уровень уверенности модели.

В области XR_WRIST предсказания преимущественно уверенные ($prob > 0.8$), что отражает хорошую разделимость классов; единичные ошибки носят пограничный характер (например, $prob \approx 0.42$) и потенциально компенсируются агрегацией на уровне исследования. Для XR_ELBOW характерны FN вблизи порога ($prob \approx 0.34-0.48$), что указывает на слабый сигнал на части снимков и чувствительность итоговых метрик к выбранному порогу. Наиболее сложные

области - XR_FINGER и XR_HAND - демонстрируют одновременно уверенные FP (prob≈0.7-0.85 при GT=0) и FN, что указывает на влияние артефактов и на ограничение модели в локализации мелких структур. Для XR_SHOULDER наблюдаются жесткие FN (prob≈0.20-0.39 при GT=1), что согласуется с высокой вариативностью проекций и менее выраженными локальными признаками патологии. В XR_HUMERUS встречаются FP с prob≈0.55-0.60, что может быть связано с ложными коррелятами (высококонтрастные элементы укладки/фиксации), а также с неоднородностью данных. Поэтому, qualitative-примеры подтверждают результаты количественных графиков: сильные области характеризуются высокой уверенностью и малым перекрытием распределений, тогда как проблемные области демонстрируют либо пограничные решения, либо устойчивые ложные срабатывания, требующие дальнейших улучшений.

	area	val_size_img	AUC_img	ACC_img	Precision_img	Recall_img	F1_img	AUC_study	ACC_study	Precision_study	Recall_study	F1_study	AUC_stored_best_study	TN	FP
XR_HUMERUS	288	0.9143	0.8299	0.7862	0.8929	0.8361	0.9333	0.8741	0.8289	0.9403	0.8811	0.9333	114	34	
XR_WRIST	659	0.8864	0.8270	0.9209	0.6712	0.7765	0.9094	0.8523	0.9429	0.6804	0.7904	0.9094	347	17	
XR_ELBOW	465	0.8988	0.7935	0.9467	0.6174	0.7474	0.9088	0.8038	0.9070	0.5909	0.7156	0.9088	227	8	
XR_FOREARM	301	0.8638	0.7841	0.8909	0.6490	0.7510	0.8988	0.7970	0.8936	0.6562	0.7568	0.8988	138	12	
XR_FINGER	461	0.8459	0.7766	0.8273	0.7368	0.7794	0.8609	0.7886	0.8194	0.7108	0.7613	0.8609	176	38	
XR_SHOULDER	563	0.8309	0.7851	0.8428	0.6942	0.7613	0.8499	0.7835	0.8354	0.6947	0.7586	0.8499	249	36	
XR_HAND	460	0.7967	0.7261	0.7368	0.5185	0.6087	0.8263	0.7725	0.8500	0.5152	0.6415	0.8263	236	35	
FN	TP	val_size_img_check	pos_count	neg_count	pos_ratio	best_auc_study	lr	weight_decay	mixup_alpha	cutmix_alpha	aug_p	erase_q	alpha_can	erase_every	
15	125	288	140	148	0.4861	0.9333	8.5607e-05	2.7338e-06	0.2441	0.2990	0.6663	0.8240	0.6710	6	
97	198	659	295	364	0.4476	0.9094	6.2674e-05	1.1878e-04	0.1209	0.1510	0.5125	0.7007	0.6240	10	
88	142	465	230	235	0.4946	0.9088	3.5747e-05	6.3512e-04	0.4392	0.8980	0.3780	0.7312	0.3232	12	
53	98	301	151	150	0.5017	0.8988	2.0026e-04	6.5027e-07	0.2063	0.6108	0.7609	0.8012	0.3950	9	
65	182	461	247	214	0.5358	0.8609	1.3320e-04	2.5862e-06	0.5647	1.1636	0.3066	0.7729	0.5592	10	
85	193	563	278	285	0.4938	0.8499	1.0695e-04	1.6829e-04	0.1161	1.1013	0.6576	0.8316	0.4455	9	
91	98	460	189	271	0.4109	0.8263	5.9050e-05	1.2241e-07	0.4935	0.0174	0.6265	0.8379	0.3549	11	

Рисунок 29 - MODEL 2 - FULL SUMMARY PER AREA

Таблица 20 дополняет AUC-оценку тем, что переводит ранжирование в «операционный режим» классификатора: помимо AUC здесь представлены ACC/Precision/Recall/F1 и абсолютные компоненты ошибок TN/FP/FN/TP при фиксированном пороге 0.5. Важно, что AUC характеризует качество ранжирования по всем возможным порогам, тогда как показатели Precision/Recall/F1 описывают поведение модели при конкретном выборе порога и, следовательно, позволяют понять какой тип ошибок преобладает и как именно модель проваливается в каждой анатомической области.

Во всех областях сохраняется закономерность $AUC_{study} > AUC_{img}$, однако таблица показывает, что высокий AUC не гарантирует высокий F1 при пороге 0.5: например, для XR_ELBOW и XR_WRIST при близких значениях $AUC_{study} \approx 0.909$

наблюдаются разные рабочие профили. В XR_WRIST модель демонстрирует более сбалансированный итог ($F1_{study} \approx 0.790$) за счет сочетания высокой точности и умеренной полноты ($Precision_{study} \approx 0.943$; $Recall_{study} \approx 0.680$). Для XR_ELBOW, напротив, при высокой $Precision_{study} \approx 0.907$ полнота снижается до $Recall_{study} \approx 0.591$ и $F1_{study}$ падает до ≈ 0.716 . Это означает, что модель хорошо отделяет классы в смысле ранжирования, но в жестком режиме порога 0.5 становится слишком консервативной и чаще отдает отрицательное решение, что приводит к росту пропуску патологии. Для XR_HUMERUS при высоком $AUC_{study} \approx 0.933$ профиль ошибок смещен в сторону высокой чувствительности ($Recall_{study} \approx 0.940$): доля пропусков относительно невелика по сравнению с количеством корректно найденных положительных случаев, что согласуется со сценарием скрининга, где критичнее уменьшать FN. Для XR_HAND наблюдается противоположная картина: хотя AUC_{study} остается приемлемым (≈ 0.826), $Recall_{study}$ составляет лишь ≈ 0.515 , а $F1_{study} \approx 0.642$. Это означает, что при фиксированном пороге модель пропускает значимую часть положительных исследований кисти; следовательно, проблема здесь не только в ранжировании, но и в том, что распределение вероятностей сильнее перекрывает, и стандартный порог оказывается неудачным для данной области. Области XR_FOREARM, XR_FINGER и XR_SHOULDER занимают промежуточное положение и демонстрируют компромиссные профили: $F1_{study}$ держится на уровне ≈ 0.756 - 0.761 , при этом $Precision_{study}$ и $Recall_{study}$ не экстремальны, но и не достигают лучших значений. Практически это указывает на наличие существенной доли пограничных исследований, где модель колеблется около порога. Именно такие области наиболее чувствительны к выбору *decision threshold* и к процедурам калибровки вероятностей.

В Model 2 качество следует оценивать в двух плоскостях: (1) AUC_{study} как основную метрику ранжирования на уровне исследования и (2) профиль ошибок при выбранном пороге ($Precision/Recall/F1$ и $TN/FP/FN/TP$), который отражает прикладную сторону модели.

```

AUC_img: macro=0.8624 | weighted=0.8601
ACC_img: macro=0.7889 | weighted=0.7892
Precision_img: macro=0.8502 | weighted=0.8560
Recall_img: macro=0.6829 | weighted=0.6728
F1_img: macro=0.7515 | weighted=0.7488

```

Рисунок 30(а) - GLOBAL STATS (IMAGE-LEVEL)

area	XR_HUMERUS
AUC_study	0.9333
ACC_study	0.8741
F1_study	0.8811
val_size_img	288
pos_count	140
neg_count	148

Рисунок 30(в) - BEST AREA (BY STUDY AUC)

```

AUC_study: macro=0.8839 | weighted≈0.8810
ACC_study: macro=0.8102 | weighted≈0.8092
Precision_study: macro=0.8682 | weighted≈0.8727
Recall_study: macro=0.6841 | weighted≈0.6717
F1_study: macro=0.7579 | weighted≈0.7533

```

Рисунок 30(б) - GLOBAL STATS (STUDY-LEVEL)

area	XR_HAND
AUC_study	0.8263
ACC_study	0.7725
F1_study	0.6415
val_size_img	460
pos_count	189
neg_count	271

Рисунок 30(г) - WORST AREA (BY STUDY AUC)

В дополнение к пер-областным результатам была рассчитана агрегированная статистика по всем анатомическим областям отдельно для image-level и study-level. На уровне отдельных снимков модель демонстрирует macro-AUC = 0.862 (weighted = 0.860) и macro-ACC = 0.789. При этом наблюдается характерный дисбаланс между точностью и полнотой: Precision_img остается высокой (macro = 0.850; weighted = 0.856), тогда как Recall_img заметно ниже (macro = 0.683; weighted = 0.673). Такой профиль означает, что при фиксированном пороге модель в среднем принимает более осторожные решения на отдельных кадрах: ложноположительные срабатывания ограничиваются, но часть патологий на уровне конкретного изображения не детектируется (рост FN на image-level).

На уровне исследования качество повышается: macro-AUC = 0.884 (weighted ≈ 0.881) и macro-ACC = 0.810 (weighted ≈ 0.809). Рост AUC и accuracy на study-level отражает ключевую особенность постановки MURA: метка задана на уровне исследования, а отдельные изображения внутри одного study могут быть неодинаково информативны. Это объясняет, почему при сопоставимых значениях полноты (Recall_study macro = 0.684) F1 и ACC на study-level оказываются немного выше, а AUC растет заметнее: улучшается именно ранжирование исследований относительно друг друга, что является более корректным критерием для MURA. Важно, что и на study-level сохраняется точностной характер модели: Precision_study

(macro = 0.868; weighted \approx 0.873) остается существенно выше Recall_study (macro = 0.684; weighted \approx 0.672). Следовательно, при пороге 0.5 модель чаще ошибается пропуском патологии, чем ложным срабатыванием, а оптимальный рабочий порог может зависеть от выбранного клинического сценария.

Лучшая анатомическая область по метрике study-level AUC - XR_HUMERUS (AUC_study = 0.933; ACC_study = 0.874; F1_study = 0.881 при размере валидации 288, pos/neg = 140/148). Модель не только корректно ранжирует исследования плечевой кости, но и уверенно работает при фиксированном пороге, сохраняя баланс между точностью и полнотой. Наиболее сложной областью остается XR_HAND (AUC_study = 0.826; ACC_study = 0.773; F1_study = 0.642 при val_size 460, pos/neg = 189/271). Здесь снижение F1 при умеренной accuracy показывает, что задача осложняется сильной внутриклассовой вариативностью и перекрытием распределений. В таком режиме AUC остается выше случайного уровня, то есть ранжирование всё ещё работает.

В целом глобальная статистика подтверждает два системных вывода: (1) study-level является более информативным уровнем оценки для MURA и демонстрирует ожидаемое улучшение относительно image-level за счет агрегации снимков; (2) модель характеризуется повышенной точностью при более умеренной полноте, поэтому дальнейшее улучшение качества для трудных областей логично искать либо через настройку порога/калибровку вероятностей, либо через усиление модели на уровне локализации и работы с мелкими деталями.

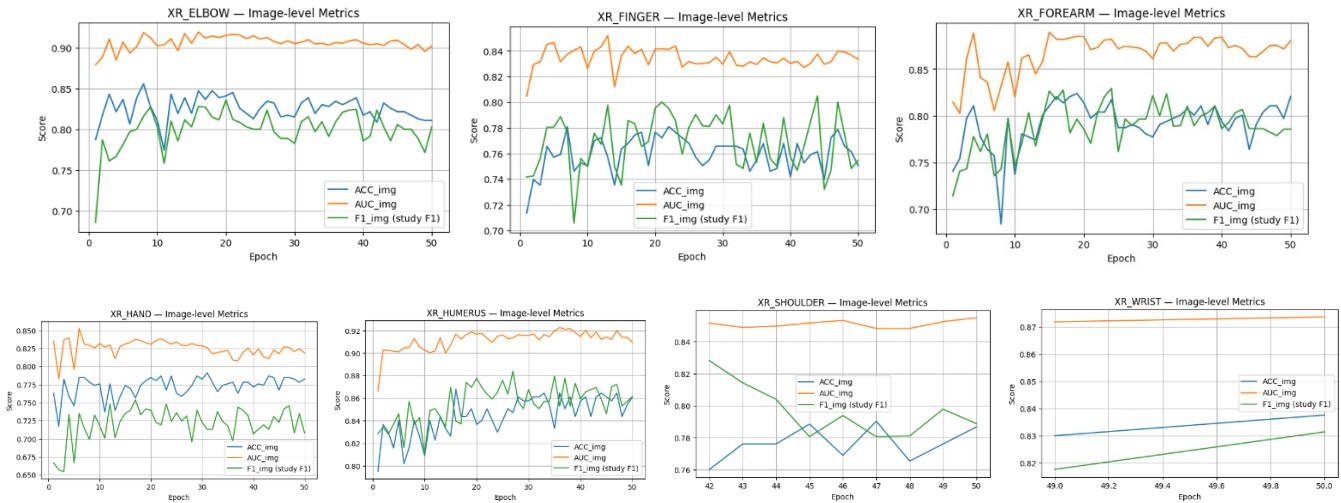


Рисунок 31 - Image-level Metrics

На графиках показана динамика метрик на уровне отдельных снимков: ACC_img, AUC_img и F1_img по эпохам. Эти кривые используются для контроля обучения, потому что модель предсказывает вероятность для каждого изображения и именно на этом уровне считается лосс. AUC_img обычно стабилизируется раньше и лучше отражает качество ранжирования вероятностей, тогда как ACC_img и F1_img сильнее колеблются, поскольку зависят от выбранного порога (0.5) и распределения ошибок FP/FN.

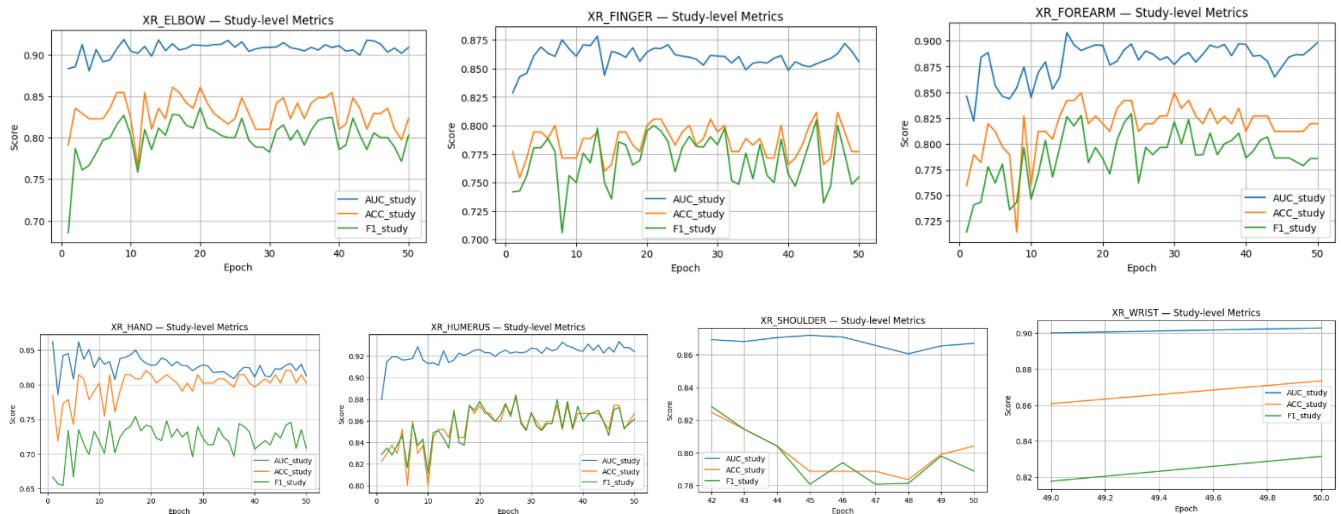


Рисунок 32 - Study-level Metrics

На графиках показана динамика study-level метрик (AUC_study, ACC_study, F1_study) по эпохам - это итоговая оценка по исследованию после агрегации предсказаний по нескольким снимкам. AUC_study быстро выходит на плато и

остается относительно стабильной (устойчивое ранжирование), тогда как ACC_study и особенно F1_study колеблются сильнее, потому что зависят от фиксированного порога и баланса FP/FN. Наиболее стабильные кривые у XR_HUMERUS и группы XR_WRIST / XR_ELBOW / XR_FOREARM, а у XR_FINGER / XR_HAND заметна большая пилообразность F1 - признак перекрытия классов и чувствительности к выбору порога.

Результаты финальной DenseNet

area	val_size_img	AUC_img	ACC_img	Precision_img	Recall_img	F1_img	AUC_study	ACC_study	Precision_study	Recall_study	F1_study
XR_HUMERUS	288	0.9200	0.8646	0.8531	0.8714	0.8622	0.9333	0.8741	0.8788	0.8657	0.8722
XR_WRIST	659	0.9050	0.8452	0.8434	0.8034	0.8229	0.9214	0.8692	0.8837	0.7835	0.8306
XR_ELBOW	465	0.9125	0.8280	0.8099	0.8522	0.8305	0.9181	0.8608	0.8333	0.8333	0.8333
XR_FOREARM	301	0.8893	0.8106	0.8507	0.7550	0.8000	0.9076	0.8421	0.8772	0.7812	0.8264
XR_FINGER	461	0.8735	0.7852	0.7960	0.8057	0.8008	0.8941	0.8114	0.7907	0.8193	0.8047
XR_SHOULDER	563	0.8627	0.7904	0.7985	0.7698	0.7839	0.8893	0.8093	0.8222	0.7789	0.8000
XR_HAND	460	0.8352	0.7609	0.8015	0.5556	0.6562	0.8624	0.7844	0.8571	0.5455	0.6667

Рисунок 33 - MODEL 3 (FINAL DENSENET) - METRICS (image + study)

Модель 3 демонстрирует наиболее ровный и клинически пригодный профиль качества на уровне исследований. По всем анатомическим зонам AUC_study находится в диапазоне 0.862-0.933, причем сильнее всего модель работает на XR_HUMERUS (AUC_study=0.933) и XR_WRIST (0.921), где сохраняется высокая точность ранжирования и хороший баланс ошибок. Для группы XR_ELBOW / XR_FOREARM AUC_study также остается высоким (0.918 и 0.908), а рост F1_study указывает, что модель не просто правильно сортирует исследования, но и уверенно отделяет классы при фиксированном пороге. На более сложных областях (XR_FINGER, XR_SHOULDER) качество остается стабильным (AUC_study≈0.889-0.894) при заметно более высоком Recall_study (~0.78-0.82), что важно для задач скрининга (меньше пропусков патологий). Самая проблемная зона по-прежнему XR_HAND (AUC_study=0.862): при высокой precision модель теряет recall (Recall_study=0.5455), то есть часть патологий остается ниже порога - обычная ситуация для кисти из-за мелких структур, наложений и артефактов. В целом Model 3 улучшает устойчивость и баланс метрик по областям: качество сохраняется высоким не только за счет AUC, но и за счет более согласованных ACC/F1 на уровне исследования.

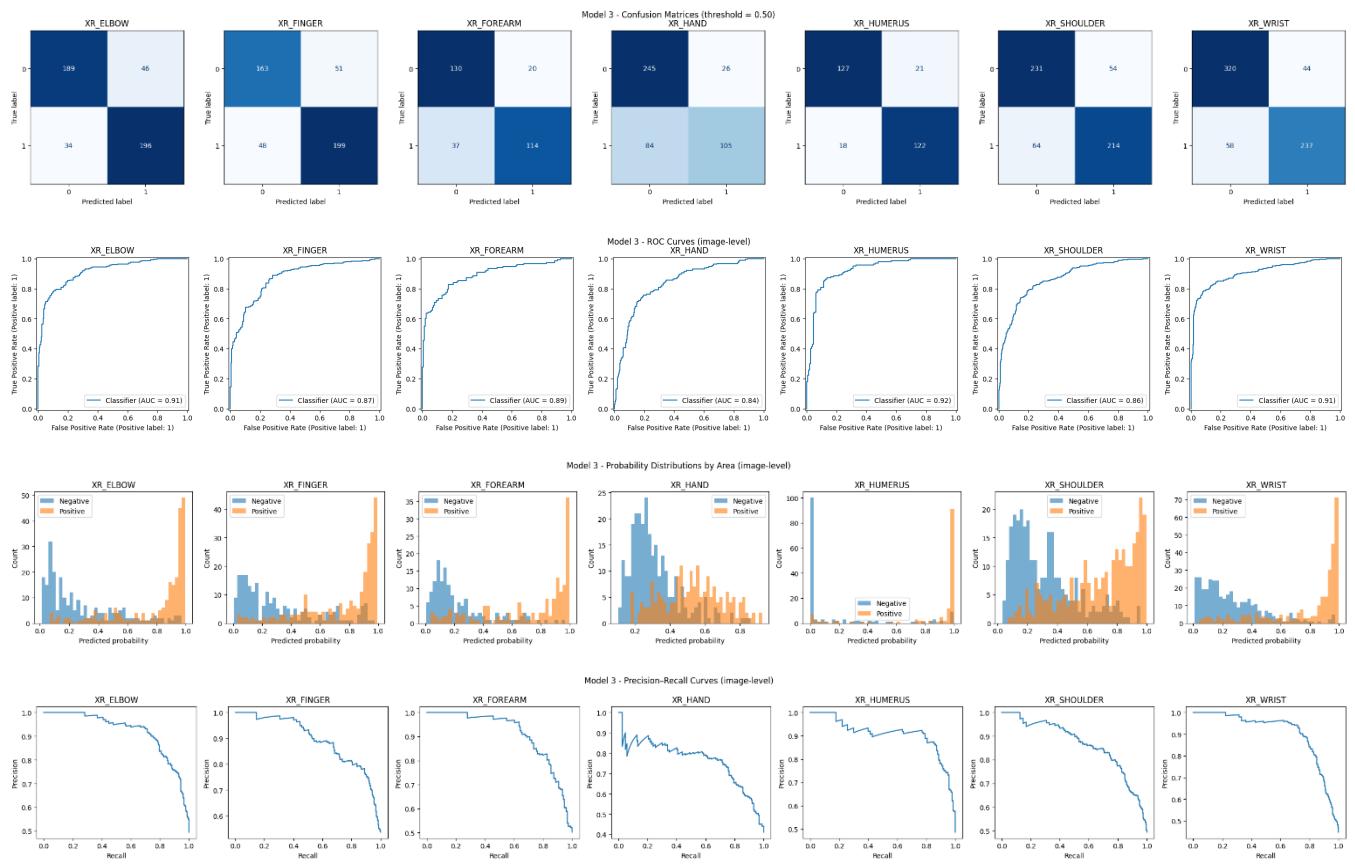


Рисунок 34 - Confusion Matrices, ROC Curves, Probability Distributions by Area, Precision–Recall Curves

По матрицам ошибок при пороге 0.5 видно, что для сильных областей (XR_WRIST, XR_ELBOW, XR_HUMERUS, XR_FOREARM) модель формирует устойчивое разделение классов. Для XR_FINGER и XR_SHOULDER сохраняется больше пограничных случаев, что проявляется в большем числе ошибок при фиксированном пороге и указывает на высокую вариативность укладок и тонкость диагностических признаков в этих областях. Наиболее сложной остается XR_HAND: даже при улучшении качества ранжирования эта область характеризуется высокой долей неоднозначных изображений, поэтому при пороге 0.5 сохраняется заметное число ошибок. ROC-кривые (image-level) подтверждают общую картину: для XR_HUMERUS, XR_WRIST и XR_ELBOW кривые приближены к верхнему левому углу, что соответствует высокому AUC и устойчивому ранжированию, тогда как для XR_HAND и частично XR_SHOULDER наблюдается более пологая форма, отражающая частое пересечение оценок нормы и патологии. Гистограммы вероятностей (image-level) дополнительно объясняют природу ошибок: в сильных областях распределения имеют выраженную

двуходмодальность (норма смещена к низким вероятностям, патология - к высоким), и зона перекрытия относительно узкая, что приводит к более уверененным решениям; в XR_FINGER и XR_SHOULDER перекрытие заметно шире, поэтому увеличивается доля borderline-предсказаний; в XR_HAND перекрытие наиболее выражено, что делает результаты чувствительными к выбору порога и ухудшает стабильность при попытке одновременно максимизировать recall и сохранить precision. Анализ PR-кривых показывает, что в сильных областях можно повышать полноту без резкого падения точности, тогда как для XR_HAND рост recall сопровождается более быстрым снижением precision, что указывает на риск лавины ложноположительных срабатываний при агрессивном пороге. В целом, визуальный анализ графиков согласуется с численными метриками: Model 3 демонстрирует более уверенное разделение классов и более стабильное ранжирование в большинстве областей, а основные ограничения сосредоточены в доменах с высокой структурной сложностью и вариативностью (прежде всего XR_HAND), где целесообразны отдельная настройка порога и/или методы, лучше работающие с локализацией и контекстом.

Approximate Average Precision (AP) per area:

XR_ELBOW: AP ≈ 0.9196

XR_FINGER: AP ≈ 0.8943

XR_FOREARM: AP ≈ 0.9093

XR_HAND: AP ≈ 0.7641

XR_HUMERUS: AP ≈ 0.9066

XR_SHOULDER: AP ≈ 0.8664

XR_WRIST: AP ≈ 0.9073

Для дополнительной оценки качества на несбалансированных пороговых режимах была рассчитана Average Precision (AP)¹¹, соответствующая площади под precision-recall кривой. Наилучшие значения получены для XR_ELBOW (≈ 0.920) и

¹¹ AP (Average Precision) — это площадь под precision-recall кривой, то есть метрика, которая сильнее «наказывает» модель за ложноположительные при попытке поднять полноту. ROC-AUC оценивает качество ранжирования в целом, а AP показывает, насколько «чистыми» остаются предсказания.

группы XR_FOREARM/XR_WRIST/XR_HUMERUS (≈ 0.907 - 0.909), что указывает на устойчивое сохранение точности при увеличении полноты. Наиболее сложной областью остаётся XR_HAND (AP ≈ 0.764), где рост recall сопровождается быстрым падением precision, что отражает выраженное перекрытие распределений вероятностей и повышенный уровень ложноположительных срабатываний.

```
XR_ELBOW 465 465 min_prob= 0.011838005855679512 max_prob= 0.996427595615387 nan_prob= 0  
XR_FINGER 461 461 min_prob= 0.003868815954774618 max_prob= 0.9952248334884644 nan_prob= 0  
XR_FOREARM 301 301 min_prob= 0.004420446697622538 max_prob= 0.9997703433036804 nan_prob= 0  
XR_HAND 460 460 min_prob= 0.10197984427213669 max_prob= 0.9254005551338196 nan_prob= 0  
XR_HUMERUS 288 288 min_prob= 9.473026148043573e-05 max_prob= 0.9999266862869263 nan_prob= 0  
XR_SHOULDER 563 563 min_prob= 0.030970672145485878 max_prob= 0.9960456490516663 nan_prob= 0  
XR_WRIST 659 659 min_prob= 0.009090949781239033 max_prob= 0.994091808795929 nan_prob= 0
```

Дополнительно была проанализирована динамика выходных вероятностей sigmoid на валидационной выборке (минимальные и максимальные значения предсказаний по областям). Для большинства анатомических зон диапазон вероятностей близок к [0,1], что указывает на наличие как уверенно-негативных, так и уверенно-позитивных примеров (например, XR_HUMERUS: min $\approx 9.5e-5$, max ≈ 0.9999 ; XR_FOREARM: min ≈ 0.0044 , max ≈ 0.9998). В отличие от них, область XR_HAND демонстрирует заметно более узкий диапазон (min ≈ 0.102 , max ≈ 0.925), то есть модель редко выдает экстремально увереные решения. Такое сжатие распределения вероятностей отражает повышенную сложность данных кисти и слабую разделимость классов, что согласуется с более низкими значениями AP/AUC и F1 для XR_HAND по сравнению с остальными областями.

CALIBRATION (IMAGE-LEVEL)

XR_ELBOW: ECE $\approx 33.90\%$

XR_FINGER: ECE $\approx 31.11\%$

XR_FOREARM: ECE $\approx 35.71\%$

XR_HAND: ECE $\approx 37.49\%$

XR_HUMERUS: ECE $\approx 46.21\%$

XR_SHOULDER: ECE $\approx 29.88\%$

XR_WRIST: ECE $\approx 37.76\%$

Калибровка выходных вероятностей была оценена с использованием метрики Expected Calibration Error (ECE)¹² на image-level. Для всех областей получены высокие значения ECE ($\approx 29.9\text{-}46.2\%$), что указывает на выраженную некалиброванность предсказанных вероятностей: при хорошем качестве ранжирования (ROC-AUC/AP) сами значения sigmoid нельзя напрямую интерпретировать как вероятности заболевания. Наиболее выраженная некалиброванность наблюдается для XR_HUMERUS (ECE $\approx 46.2\%$), тогда как минимальные значения ECE отмечены для XR_SHOULDER ($\approx 29.9\%$) и XR_FINGER ($\approx 31.1\%$). Это говорит о том, что для клинически интерпретируемого вывода целесообразно применять методы пост-калибровки и подбирать пороги по study-level.

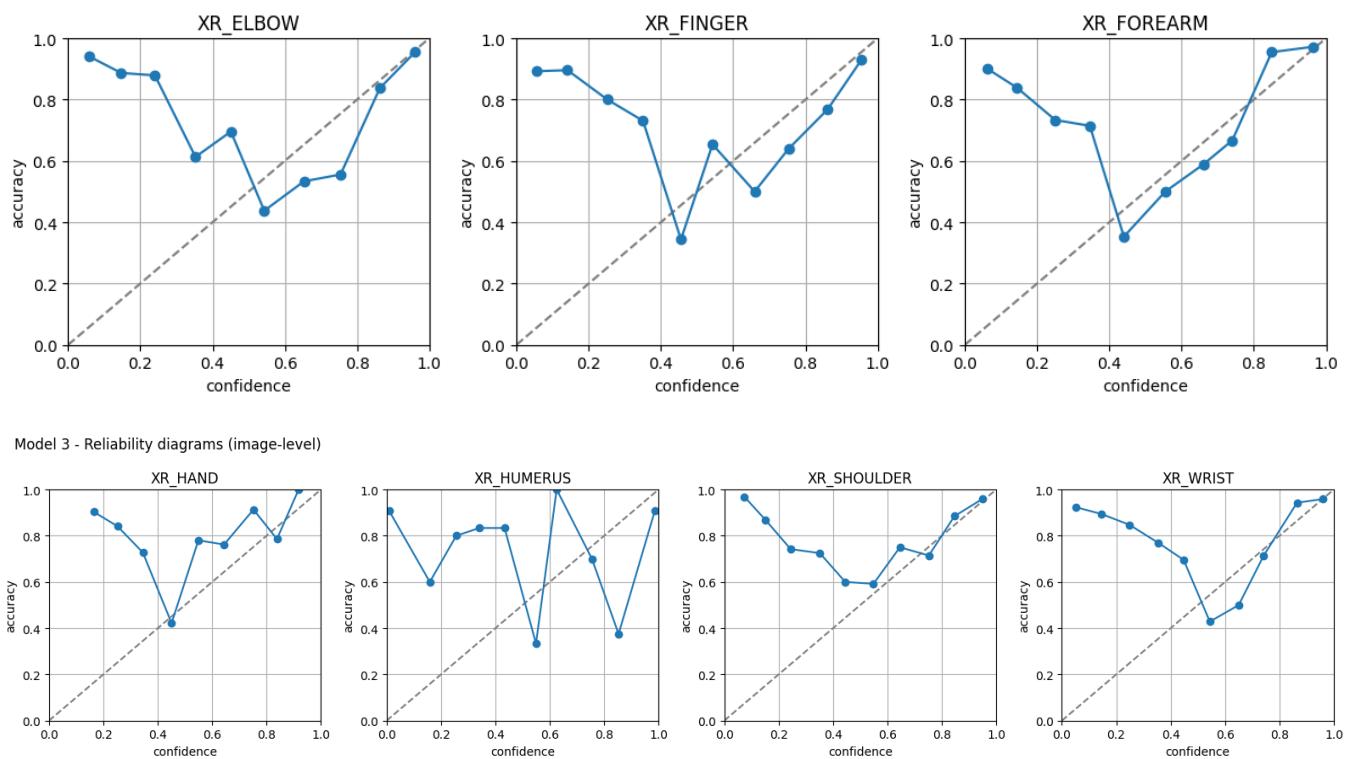


Рисунок 35 - Reliability diagrams

Для Model 3 дополнительно оценивалась калибровка предсказанных вероятностей на уровне отдельных изображений с помощью reliability diagram: по

¹² ECE (Expected Calibration Error) показывает, насколько «вероятность = частота» выполняется на практике. Если модель говорит 0.8, то из таких случаев примерно 80% должны быть положительными. Чем выше ECE, тем сильнее расхождение.

оси x отложена заявленная моделью уверенность (confidence), по оси y - фактическая доля правильных ответов (accuracy) в соответствующем диапазоне вероятностей. Идеально откалиброванная модель лежит вдоль диагонали $y = x$.

По всем областям на графиках заметен типичный U-образный профиль:

- в зоне низких confidence ($\approx 0\text{-}0.2$) кривая выше диагонали \rightarrow модель скорее недоуверена (0.1-0.2);
- в зоне средних confidence ($\approx 0.4\text{-}0.6$) у многих областей наблюдается провал ниже диагонали \rightarrow модель переоценивает себя на пограничных примерах: при уверенности около 0.5 реальная точность существенно ниже. Это соответствует наиболее сложным случаям и возможной неоднозначности/шума разметки;
- в зоне высоких confidence (≥ 0.85) кривая снова приближается к диагонали \rightarrow самые уверенные предсказания в целом более надежны.

Это согласуется с рассчитанным Expected Calibration Error (ECE): во всех областях калибровка остается слабой (ECE порядка $\sim 30\text{-}46\%$), причем наиболее выраженная нестабильность видна у XR_HUMERUS ($ECE \approx 46\%$) и в целом у областей с более неровной кривой (резкие скачки по confidence-бинам). Такая зубчатость также частично объясняется тем, что в отдельных бинах может быть мало примеров, поэтому оценка accuracy получается шумной.

Практический вывод: вероятности Model 3 стоит интерпретировать прежде всего как ранжирование риска, а не как честные проценты, и для клинически корректного использования логично добавить посткалибровку и/или отдельно подбирать пороги под нужный баланс ошибок.

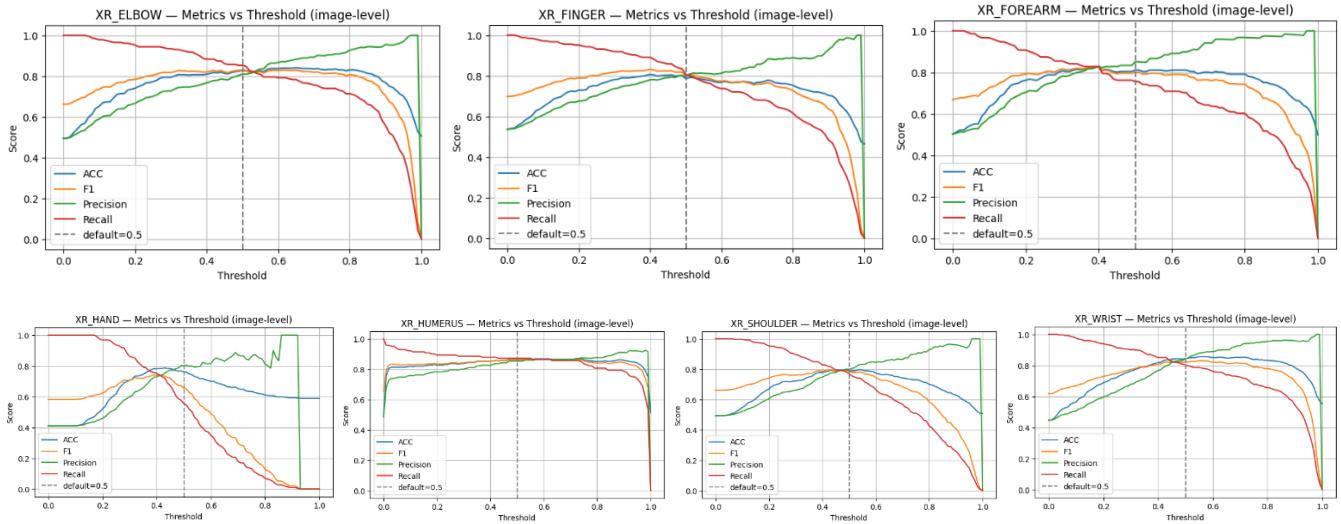
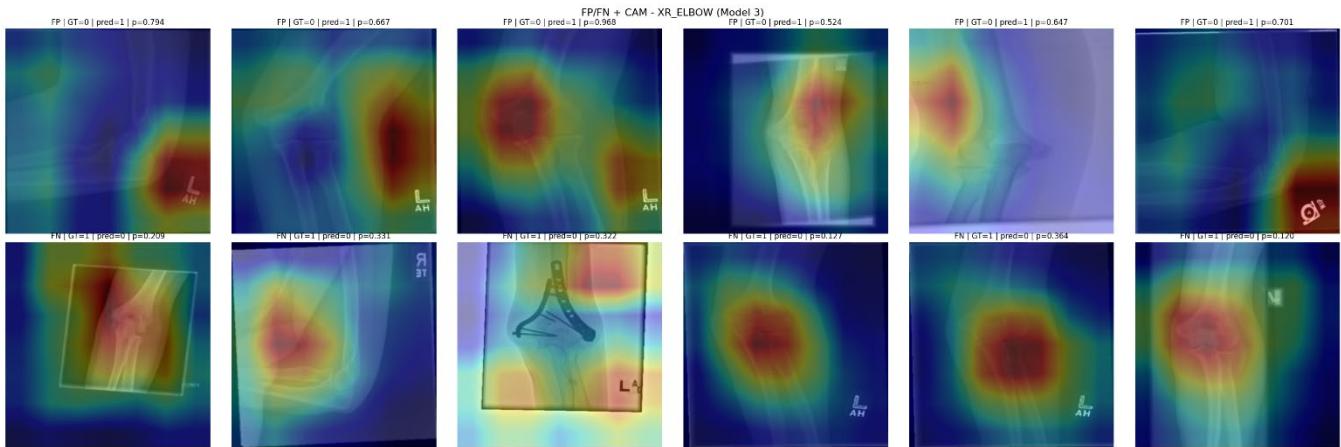
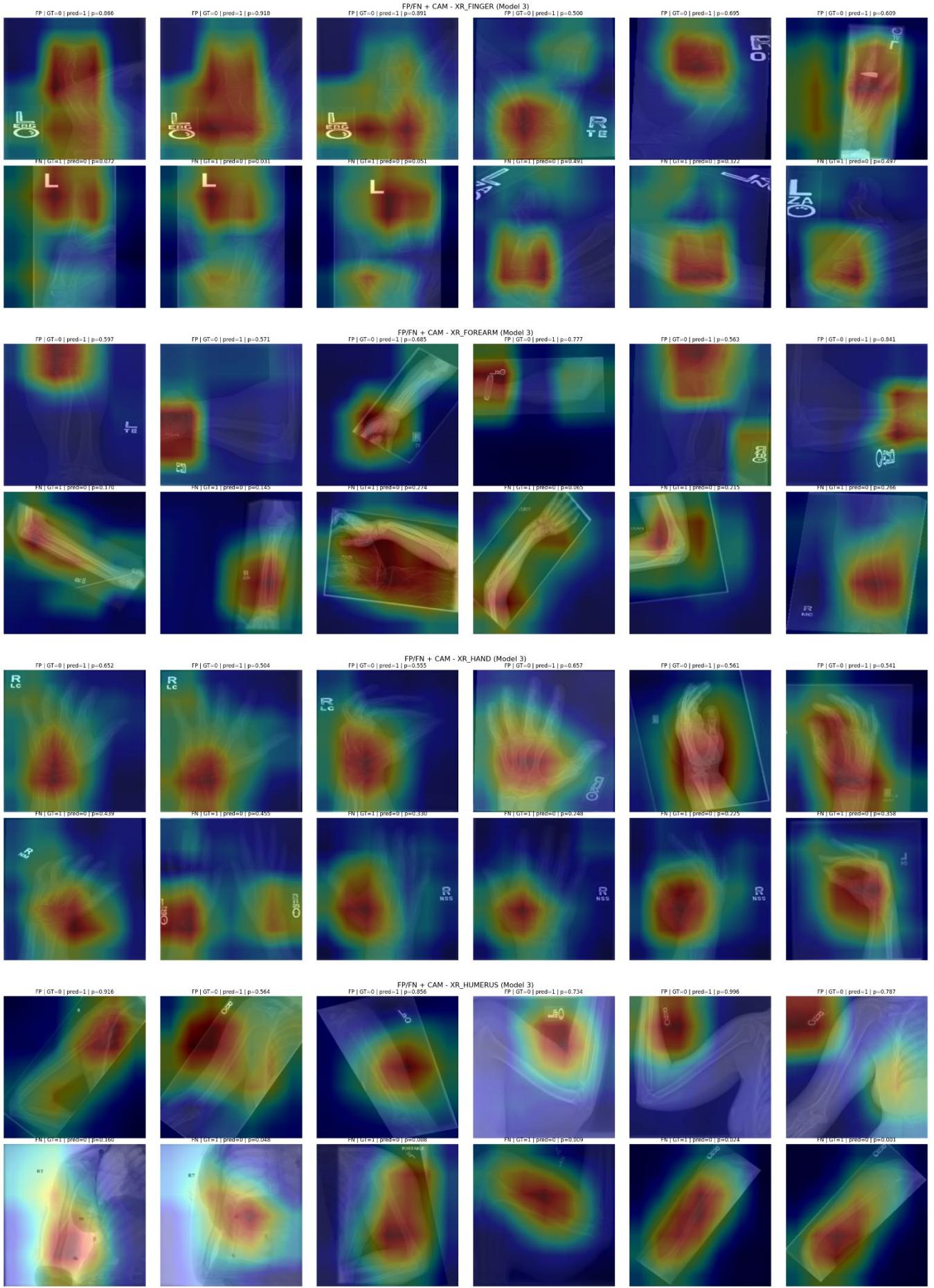


Рисунок 36 - Metrics vs Threshold

Графики метрик от порога показывают компромисс: при росте threshold precision растёт, а recall падает, и максимум F1 обычно достигается не обязательно при 0.5. Для сложных областей (особенно XR_HAND) оптимальный порог по F1 может смещаться, поэтому фиксированный threshold=0.5 дает недобор recall и много FN - это согласуется и с confusion matrices, и с перекрытием распределений вероятностей.





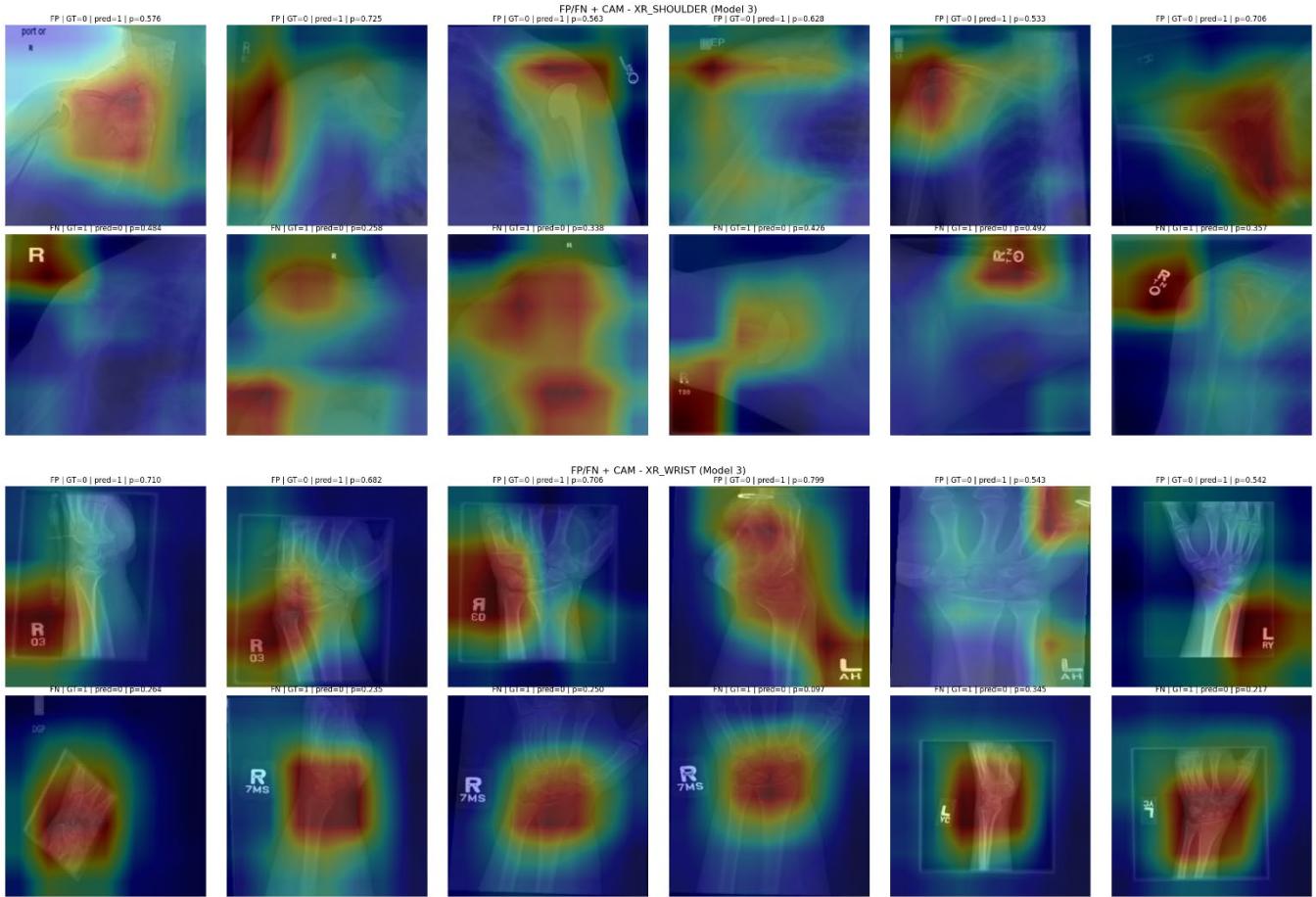


Рисунок 37 - FP/FN GALLERY WITH CAM

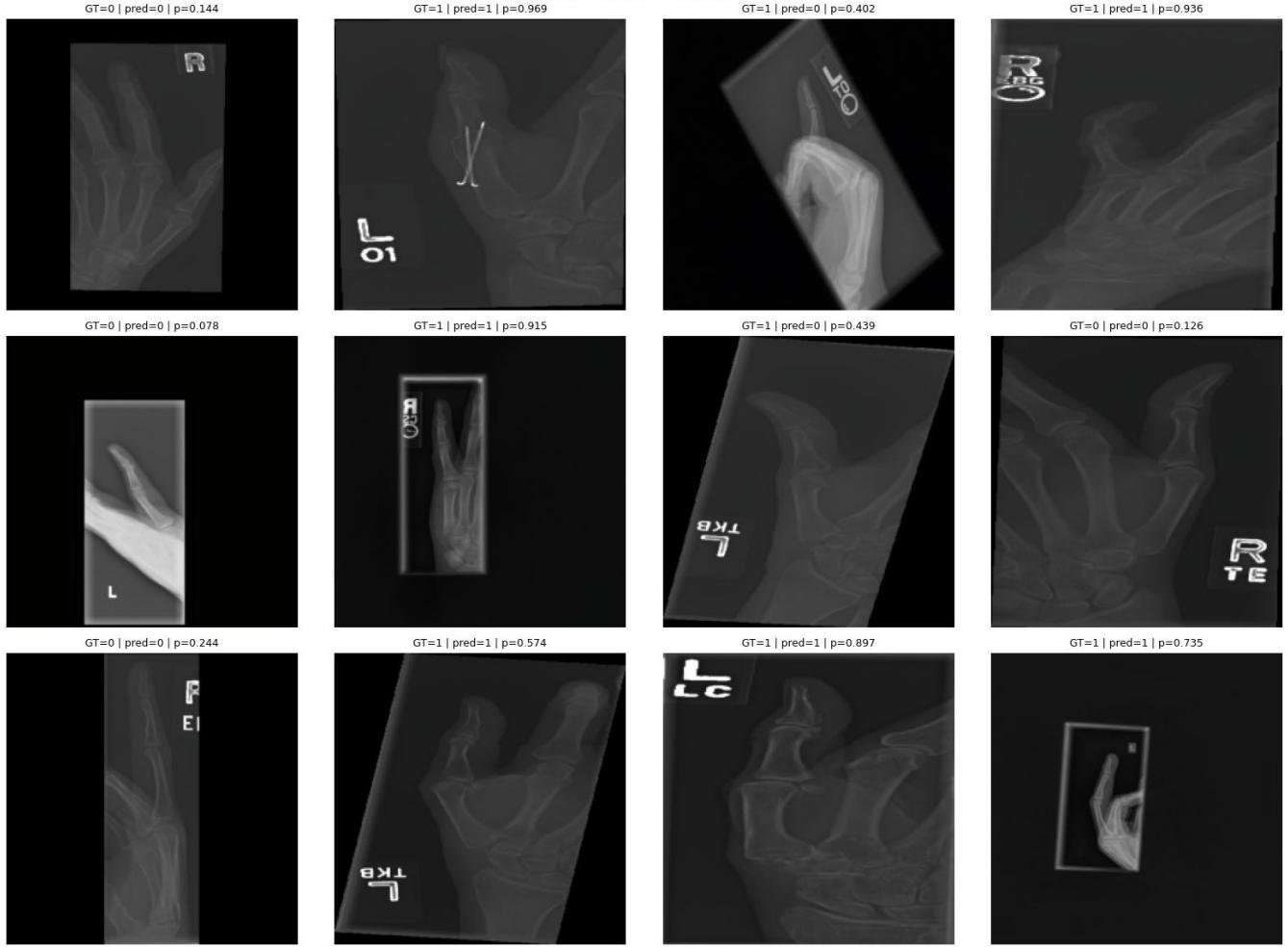
Для модели 3 дополнительно был выполнен разбор ошибок и интерпретация решений с помощью Grad-CAM. Цель этого шага - не только зафиксировать итоговые метрики, но и понять, почему модель ошибается: какие визуальные признаки реально влияют на предсказание и не использует ли сеть shortcut features, например марковку снимка или границы изображения. Такой анализ важен для медицинской задачи, потому что высокий AUC сам по себе ещё не гарантирует корректное поведение модели: при систематическом внимании к артефактам качество может ухудшаться при смене оборудования, протоколов съемки и оформления снимков.

Визуализация FP/FN через Grad-CAM показывает характерные причины ошибок. Для ложноположительных случаев (FP) внимание модели нередко смещается на неанатомические высококонтрастные элементы: метки L/R, текст, рамку коллимации, края кадра и другие артефакты. Такие признаки легко «цепляются» сетью и могут коррелировать с классом в обучающей выборке, из-за

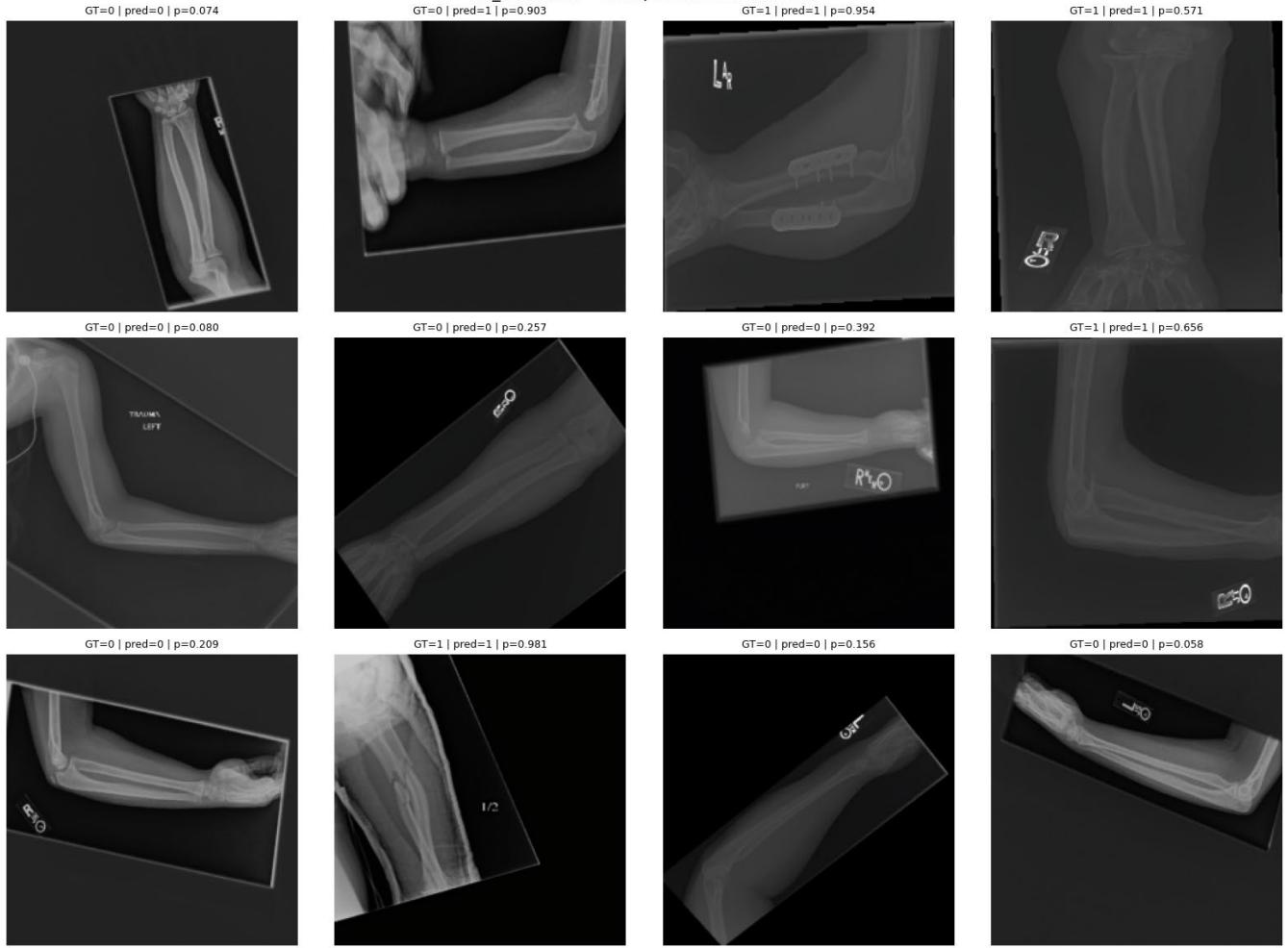
чего модель выдает уверенные срабатывания при отсутствии патологии. Для ложноотрицательных (FN) чаще наблюдается обратная ситуация: внимание распределено широко или фиксируется на крупных структурах, тогда как потенциальная патология представлена тонкими линиями/локальными нарушениями контура и теряется на фоне наложений и вариативности укладки (особенно характерно для кисти и пальцев). Получается, SAM-анализ подтверждает, что ограничения качества связаны не только с «силой» классификатора, но и с наличием артефактов и сложностью локальных признаков, что задает направления для дальнейшего улучшения (маскирование/кроп меток и рамок, аугментации против текстовых артефактов, настройка порога и упор на study-level агрегацию).



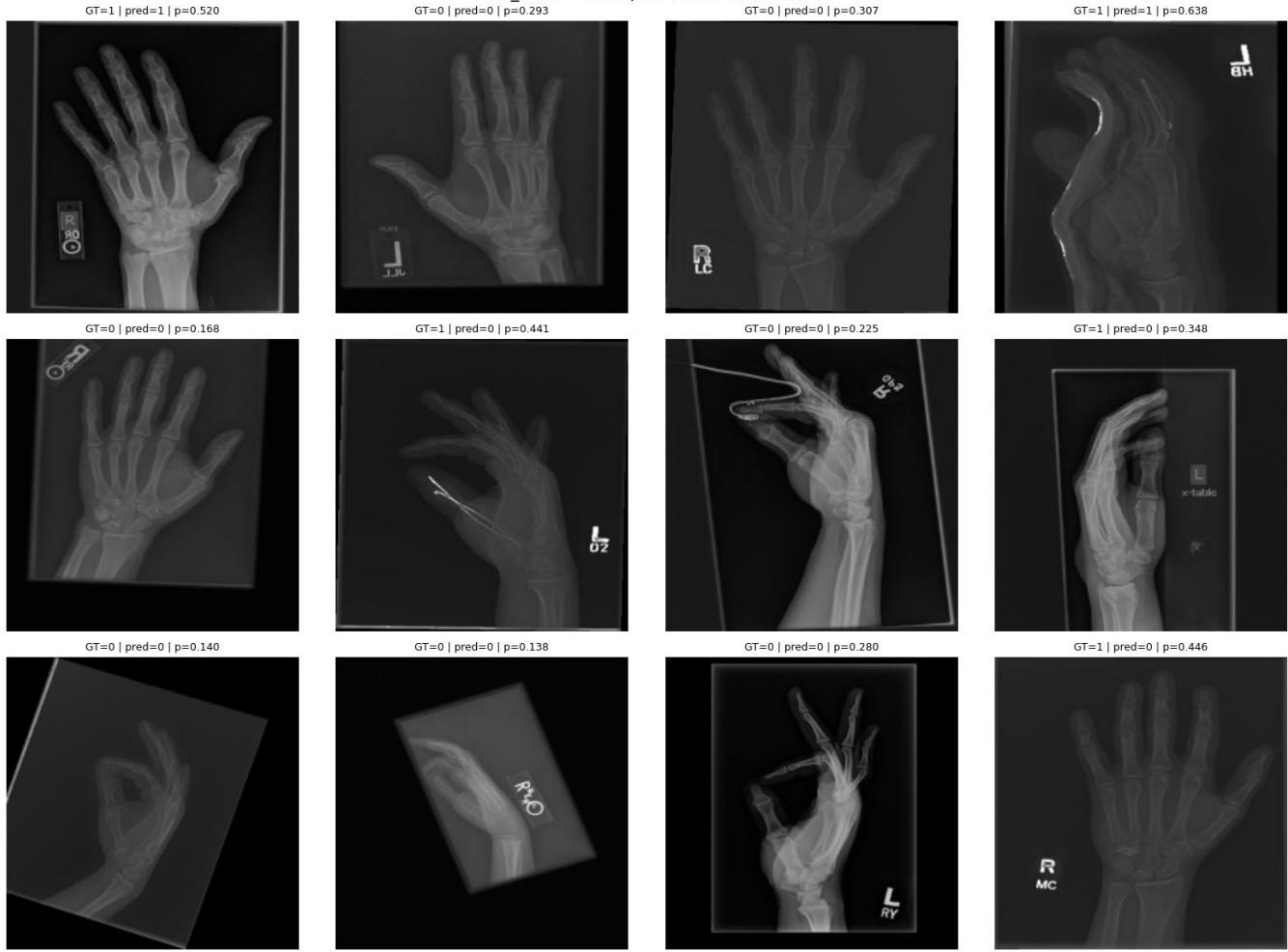
XR_FINGER - examples (Model 3, mixed)



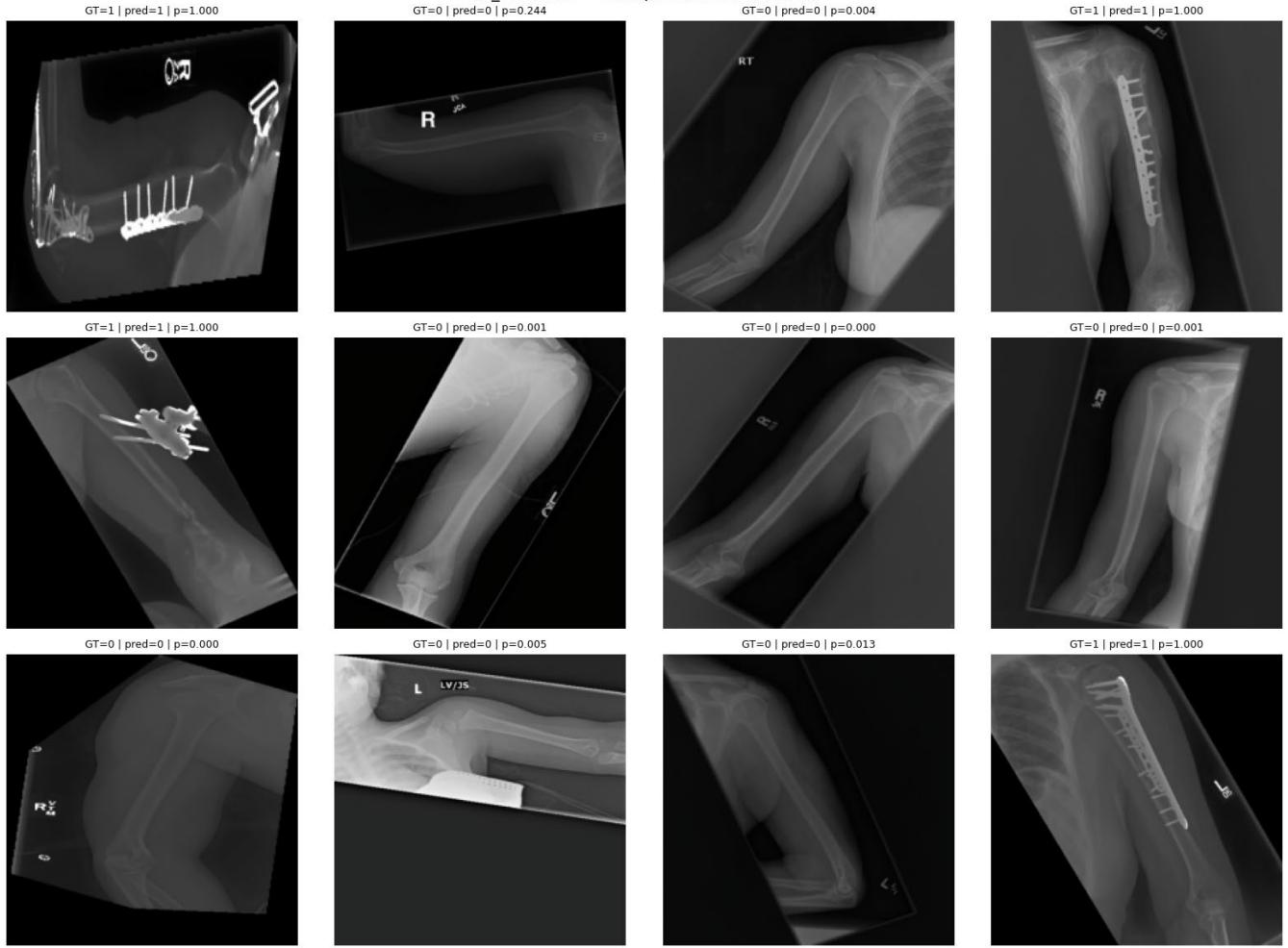
XR_FOREARM - examples (Model 3, mixed)



XR_HAND - examples (Model 3, mixed)



XR_HUMERUS - examples (Model 3, mixed)



XR_SHOULDER - examples (Model 3, mixed)

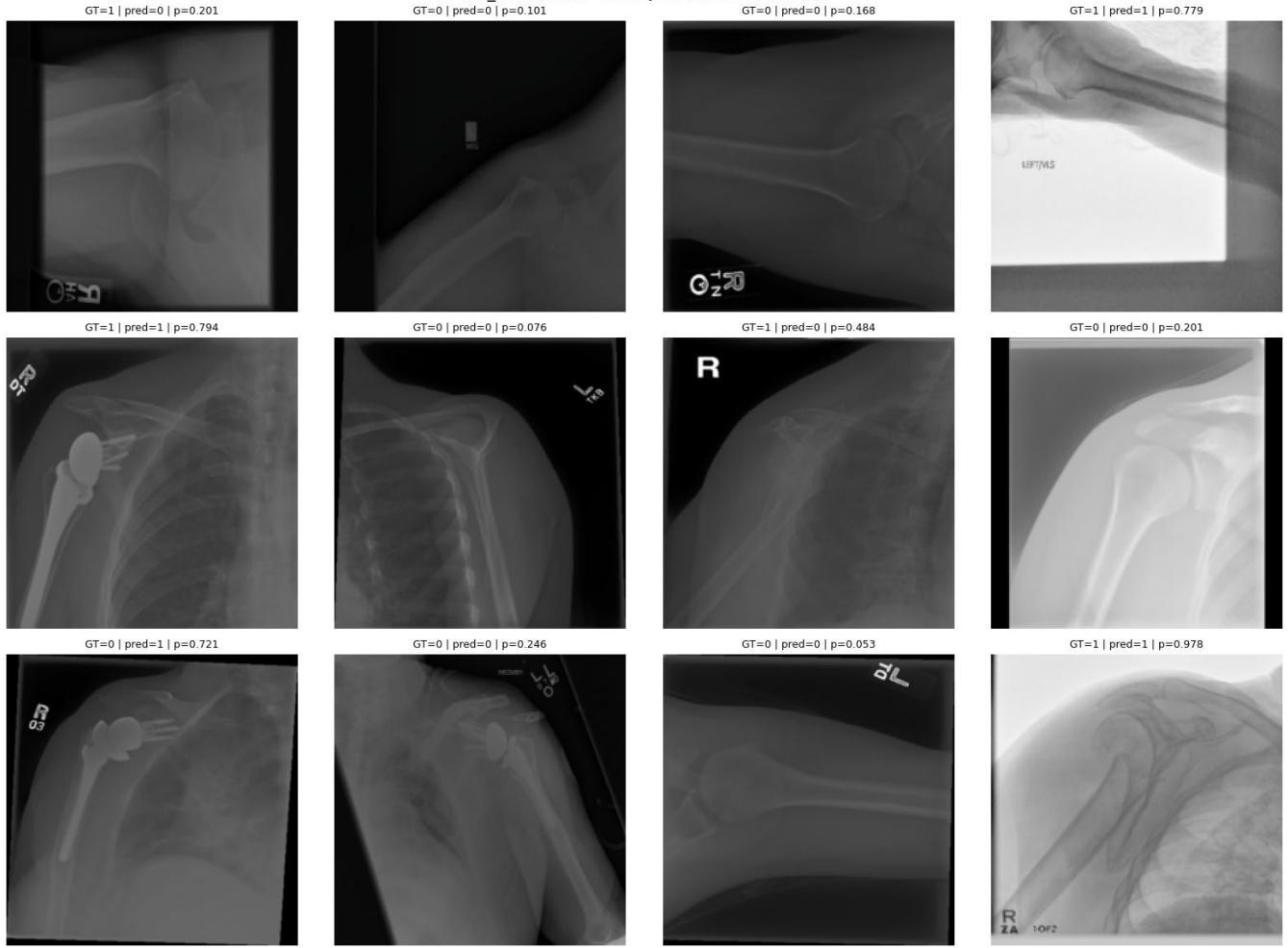




Рисунок 38 - 3 "XR_ELBOW", "XR_FINGER", "XR_FOREARM", "XR_HAND", "XR_HUMERUS", "XR_SHOULDER", "XR_WRIST"

По примерам с GT/pred/p видно, что поведение модели заметно отличается по анатомическим областям. Для XR_ELBOW в выборке примеров встречается выраженная доля ложноположительных с достаточно высокой уверенностью ($GT=0 \rightarrow pred=1$ при $p \approx 0.58-0.76$), что указывает на склонность модели «перестраховываться» и отмечать патологию на части отрицательных исследований. Для XR_FINGER картина смешанная: есть очень уверенные истинноположительные ($GT=1 \rightarrow pred=1$ при $p \approx 0.92-0.97$), но при этом встречаются ложноотрицательные с вероятностями около $p \approx 0.40-0.44$ - это пограничные случаи, где модель «почти» выбирает положительный класс, но остается ниже порога 0.5. Для XR_FOREARM в примерах заметны как сильные истинноположительные решения ($p \approx 0.95-0.98$), так и уверенные ложноположительные ($GT=0 \rightarrow pred=1$ при $p \approx 0.90$), что говорит о том, что часть отрицательных исследований содержит паттерны, которые модель

интерпретирует как патологические с высокой уверенностью; ложноотрицательные встречаются реже и обычно соответствуют низко-средним значениям p ($\approx 0.15-0.27$). Для XR_HAND наблюдается наименее устойчивое разделение классов: у положительных и отрицательных примеров часто встречаются вероятности вблизи границы решения; в частности, ложноотрицательные случаи имеют $p \approx 0.35-0.45$ (модель «склоняется» к положительному, но порог 0.5 приводит к отрицательному решению). Для XR_HUMERUS, напротив, в предоставленных примерах видно максимально «поляризованное» поведение: истинноположительные решения сопровождаются $p \approx 1.00$, а истинноотрицательные - $p \approx 0.00-0.01$, то есть модель уверенно различает классы и выдает крайние вероятности. Для XR_SHOULDER встречаются ложноположительные с заметной уверенностью ($GT=0 \rightarrow pred=1$ при $p \approx 0.72-0.78$), а также ложноотрицательные ($GT=1 \rightarrow pred=0$) с $p \approx 0.20-0.48$, что соответствует ошибкам в зоне средней/граничной уверенности. Для XR_WRIST в примерах присутствуют уверенные истинноположительные ($p \approx 0.95-0.98$), однако также наблюдаются ложноположительные со средней уверенностью ($p \approx 0.64-0.67$) и ложноотрицательные с $p \approx 0.23-0.48$, что характерно для ситуаций, когда часть случаев остается близкой к порогу и решение чувствительно к выбранному threshold.

AUC_img: macro=0.8855 weighted=0.8839	AUC_study: macro=0.9038 weighted≈0.9026
ACC_img: macro=0.8121 weighted=0.8108	ACC_study: macro=0.8359 weighted≈0.8348
Precision_img: macro=0.8219 weighted=0.8193	Precision_study: macro=0.8490 weighted≈0.8473
Recall_img: macro=0.7733 weighted=0.7708	Recall_study: macro=0.7725 weighted≈0.7680
F1_img: macro=0.7938 weighted=0.7914	F1_study: macro=0.8049 weighted≈0.8016

Рисунок 39(а) -- GLOBAL STATS (IMAGE-LEVEL)

Рисунок 39(б) - GLOBAL STATS (STUDY-LEVEL)

area	XR_HUMERUS
AUC_study	0.9333
ACC_study	0.8741
F1_study	0.8722
val_size_img	288

Рисунок 39(в) - BEST AREA (BY STUDY AUC)

area	XR_HAND
AUC_study	0.8624
ACC_study	0.7844
F1_study	0.6667
val_size_img	460

Рисунок 39(г) - WORST AREA (BY STUDY AUC)

Модель 3 демонстрирует стабильное качество как на уровне отдельных изображений, так и на уровне исследования. На image-level получены AUC (macro)=0.8855 и Accuracy (macro)=0.8121, при Precision=0.8219 и Recall=0.7733, что

соответствует F1 (macro)=0.7938. Это означает, что модель хорошо разделяет классы по вероятностям (высокий AUC) и при фиксированном пороге дает сбалансированную точность и полноту без явного перекоса в сторону FP или FN. При переходе к study-level метрики дополнительно возрастают: AUC (macro)=0.9038, Accuracy (macro)=0.8359, Precision=0.8490, Recall=0.7725, F1 (macro)=0.8049.

Лучший результат среди областей достигается для XR_HUMERUS: AUC_study=0.9333, ACC_study=0.8741, F1_study=0.8722 (val_size_img=288), что указывает на высокую разделимость нормальных и патологических случаев в этой зоне. Наиболее сложной областью остается XR_HAND: AUC_study=0.8624, ACC_study=0.7844, F1_study=0.6667 (val_size_img=460), то есть при приемлемом ранжировании (AUC) качество классификации при выбранном пороге заметно ниже, что согласуется с большей визуальной вариативностью и сложностью интерпретации патологий кисти. В целом, модель 3 обеспечивает сильное обобщение (AUC > 0.88 на image-level и > 0.90 на study-level) и остается наиболее надежной версией среди рассмотренных, при этом ключевой зоной для дальнейшего улучшения является XR_HAND.

Результаты гибридной модели

	area	val_size_img	AUC_img	ACC_img	Precision_img	Recall_img	F1_img	AUC_study
XR_SHOULDER		563	0.8102	0.7336	0.7883	0.6295	0.7000	0.8235
XR_FINGER		461	0.7831	0.7223	0.7280	0.7692	0.7480	0.8080
XR_HAND		460	0.7795	0.7000	0.7383	0.4180	0.5338	0.8017
	Precision_study	Recall_study	F1_study	best_epoch_ckpt	AUC_stored_ckpt			
	0.8194	0.6211	0.7066		57			0.810
	0.7159	0.7590	0.7368		55			0.783
	0.7714	0.4091	0.5347		59			0.779

Рисунок 40 - Результаты по областям Подхода 4

EVAL: XR_FINGER
[HybridModel] d_feat=torch.Size([16, 1024, 14, 14]), s_feat=torch.Size([16, 1024, 7, 7]), combined=torch.Size([16, 2048])
EVAL: XR_SHOULDER

```
[HybridModel] d_feat=torch.Size([16, 1024, 14, 14]), s_feat=torch.Size([16, 1024, 7, 7]),  
combined=torch.Size([16, 2048])
```

EVAL: XR_HAND

```
[HybridModel] d_feat=torch.Size([16, 1024, 14, 14]), s_feat=torch.Size([16, 1024, 7, 7]),  
combined=torch.Size([16, 2048])
```

Гибридная модель (Model 4) была введена как попытка усилить качество на «трудных» анатомических областях за счет объединения комплементарных представлений. В отличие от Model 3 (финальная DenseNet), где решение принимается на основе признаков одного сверточного экстрактора, гибридная схема использует две ветви признаков и формирует общий вектор признаков путем их конкатенации (в эксперименте: d_feat размерности 1024 и s_feat размерности 1024, далее объединение до 2048). Идея подхода состоит в том, что разные представления могут быть чувствительны к различным морфологическим маркерам, и их объединение теоретически повышает устойчивость классификации на вариативных снимках.

Эксперимент с Model 4 был выполнен только для трех областей (XR_SHOULDER, XR_FINGER, XR_HAND), поскольку именно они были выбраны как проблемные/приоритетные для улучшения, а полный прогон по всем областям существенно увеличивает вычислительные затраты и время перебора конфигураций. Такой дизайн эксперимента соответствует постановке «targeted improvement»: сначала проверяется гипотеза на наиболее сложных классах, и только при подтверждении эффекта масштабирование переносится на остальные области. Однако полученные метрики показывают, что в текущей реализации Model 4 уступает финальной Model 3 на всех трех протестированных областях. Так, на уровне исследований (study-level) AUC и F1 для XR_SHOULDER составляют около 0.8235 и 0.7066, для XR_FINGER - около 0.8080 и 0.7368, для XR_HAND - около 0.8017 и 0.5347. Для сравнения, в Model 3 соответствующие показатели на этих областях выше (например, для XR_FINGER и XR_SHOULDER F1_study порядка ~0.80, а AUC_study ~0.89; для XR_HAND F1_study ~0.67 при AUC_study ~0.86). Это означает, что простое объединение признаков (concat-fusion) само по себе не

гарантирует прироста: модель может терять обобщающую способность, «запоминать» нерелевантные сигналы и ухудшать баланс ошибок.

Отдельно важно учитывать, что итоговые показатели F1 и Recall зависят не только от ранжирующей способности (AUC), но и от выбранного порога классификации. В отчете использовался стандартный порог 0.5, который является удобным для сравнения, но не обязательно оптимальным для каждой области. В частности, для сложных областей типична ситуация, когда модель становится «консервативной» и снижает долю положительных предсказаний, что приводит к росту FN и падению Recall/F1 даже при приемлемом AUC. Поэтому для корректной интерпретации результата гибридной модели необходимо отдельно анализировать профиль ошибок и поведение метрик при изменении порога.

Далее, для более предметного выяснения причин деградации качества и определения направлений улучшения, проводится детальный разбор по диагностическим визуализациям: матрицам, ROC- и PR-кривым (включая AP по областям), графикам «метрики-порог» (ACC/Precision/Recall/F1 vs threshold), а также распределениям предсказанных вероятностей по классам и диаграммам надежности (reliability diagrams) для оценки калибровки. Такой анализ позволяет локализовать источник проблемы и выбрать наиболее рациональные корректировки: подбор порога по области, калибровку вероятностей (temperature scaling/Platt), более умное слияние признаков (gated/attention fusion вместо конкатенации), а также балансировку функции потерь или стратегии сэмплирования для повышения чувствительности на положительном классе.

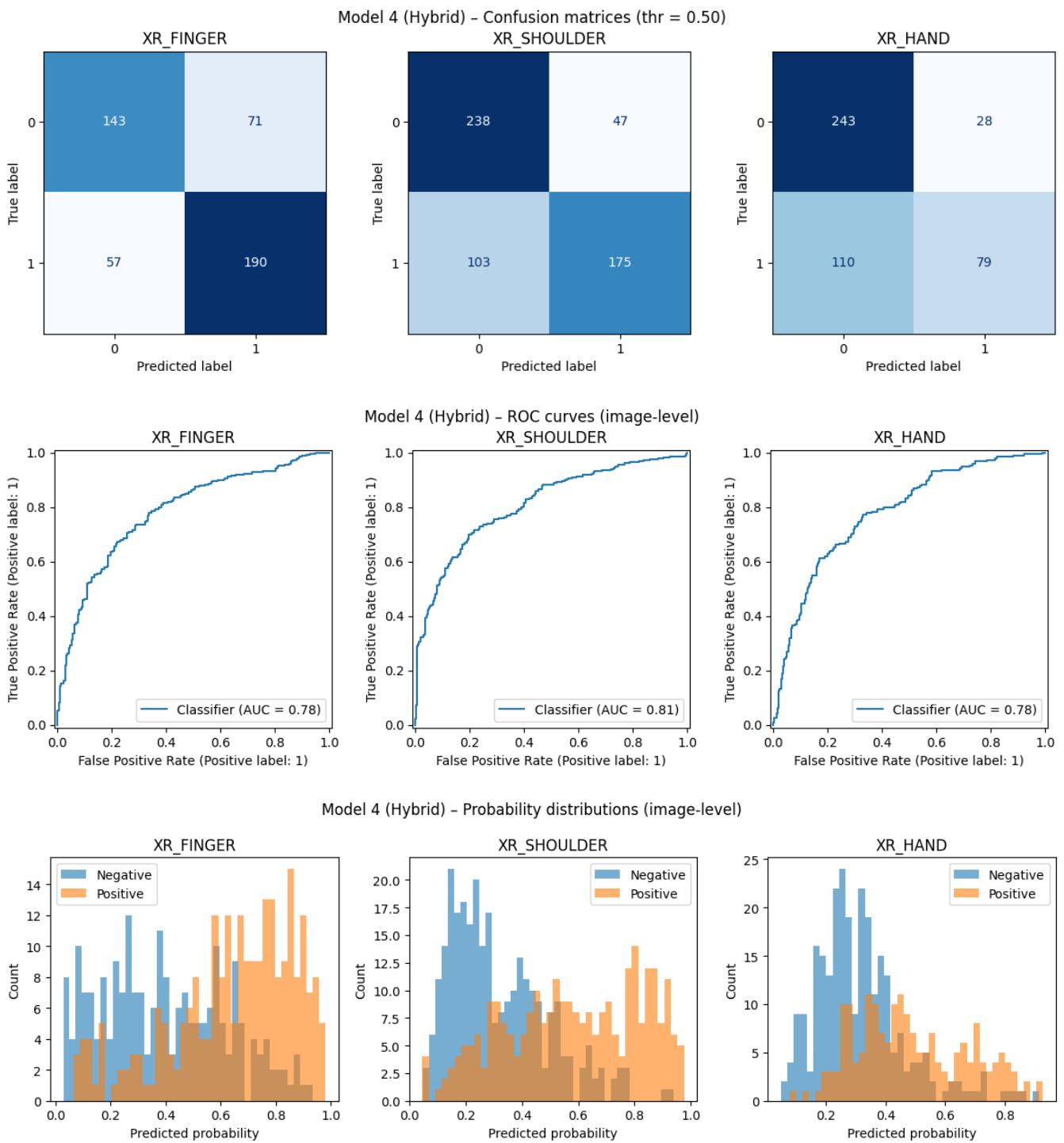


Рисунок 30 - Confusion Matrices, ROC Curves, Probability Distributions by Area

По графикам видно, что гибридная схема сейчас дает неравномерное качество по классам и особенно проседает по чувствительности на XR_HAND.

На XR_FINGER при $\text{thr}=0.5$ матрица ошибок показывает заметное число ложноположительных ($FP=71$) при одновременно высоком числе истинноположительных ($TP=190$) и умеренных пропусках ($FN=57$). Это хорошо

согласуется с ROC-кривой: $AUC \approx 0.78$ - модель ранжирует неплохо, но разделение классов не идеальное. На гистограмме вероятностей видно, что распределения positive/negative частично перекрываются в средней зоне (примерно 0.3-0.7), из-за чего при фиксированном пороге часть «нормы» уходит в патологию (рост FP), а часть патологии остается ниже порога (FN). Здесь гибрид, по сути, работает как умеренно агрессивный классификатор: лучше ловит positives, но расплачивается FP.

На XR_SHOULDER ситуация более сбалансированная по FP ($FP=47$), но заметно больше FN ($FN=103$ при $TP=175$). Это отражено и на распределениях: у positives есть смещение к большим вероятностям, но хвост positives уходит в средние/низкие значения - отсюда пропуски. ROC $AUC \approx 0.81$ подтверждает, что ранжирование лучше, чем на XR_FINGER, но выбор порога 0.5 все ещё не оптимален: модель недобирает recall (и на image-level, и на study-level recall около 0.62-0.63). Практически это означает, что для плеча полезно отдельно подобрать порог под целевую метрику (например, под F1 или под требуемую чувствительность).

На XR_HAND наиболее проблемный профиль: при $thr=0.5$ матрица ошибок показывает очень много пропусков патологии ($FN=110$ при $TP=79$) при сравнительно небольшом FP ($FP=28$). То есть модель ведет себя слишком консервативно: если уверенность не высокая, она предпочитает «0», из-за чего recall падает (≈ 0.42 на image-level и ≈ 0.41 на study-level), а F1 получается низким (≈ 0.53). Это напрямую видно и по гистограмме вероятностей: распределение positive заметно смещено вверх, но сильно пересекается с negative в диапазоне примерно 0.2-0.6, поэтому фиксированный порог 0.5 режет значимую часть истинных positives. При этом ROC $AUC \approx 0.78$ говорит, что потенциал у ранжирования есть, но порог и/или калибровка и обучение сейчас не позволяют перевести это в высокую чувствительность.

	area	val_size_img	AUC_img	ACC_img	Precision_img	Recall_img	F1_img	AUC_study	ACC_study	Precision_study	Recall_study	F1_study
XR_SHOULDER		563	0.8102	0.7336	0.7883	0.6295	0.7000	0.8235	0.7474	0.8194	0.6211	0.7066
XR_FINGER		461	0.7831	0.7223	0.7280	0.7692	0.7480	0.8080	0.7429	0.7159	0.7590	0.7368
XR_HAND		460	0.7795	0.7000	0.7383	0.4180	0.5338	0.8017	0.7186	0.7714	0.4091	0.5347
F1_study	best_epoch_ckpt		AUC_stored_ckpt	TN	FP	FN	TP	val_size_img_check	pos_count	neg_count	pos_ratio	
0.7066		57	0.8102	238	47	103	175		563	278	285	0.4938
0.7368		55	0.7831	143	71	57	190		461	247	214	0.5358
0.5347		59	0.7795	243	28	110	79		460	189	271	0.4109

Рисунок 31 - FULL SUMMARY PER AREA

Есть ли шанс обогнать Model 3? Теоретически - да, потому что гибридные модели умеют вытаскивать комплементарные признаки (например, сочетать локальные детали и более глобальный контекст). Но по текущим графикам и метрикам этот шанс пока не реализован: по этим трем областям AUC у Model 4 заметно ниже, чем у Model 3, а по XR_HAND дополнительно видно перекос в сторону низкой чувствительности. То есть сейчас гибрид - это не улучшение, а скорее диагностический эксперимент, который показал, где именно ломается пайплайн: порог 0.5 не подходит и/или признаки слияния/обучение не дают достаточного разрыва распределений.

AUC_img: macro=0.7909 weighted=0.7923	AUC_study: macro=0.8111 weighted≈0.8119
ACC_img: macro=0.7186 weighted=0.7197	ACC_study: macro=0.7363 weighted≈0.7371
Precision_img: macro=0.7515 weighted=0.7541	Precision_study: macro=0.7689 weighted≈0.7724
Recall_img: macro=0.6056 weighted=0.6073	Recall_study: macro=0.5964 weighted≈0.5982
F1_img: macro=0.6606 weighted=0.6634	F1_study: macro=0.6594 weighted≈0.6627

Рисунок 42(а) - GLOBAL STATS (IMAGE-LEVEL)

area	XR_SHOULDER
AUC_study	0.8235
ACC_study	0.7474
F1_study	0.7066
val_size_img	563
pos_count	278
neg_count	285

Рисунок 42(б) - GLOBAL STATS (STUDY-LEVEL)

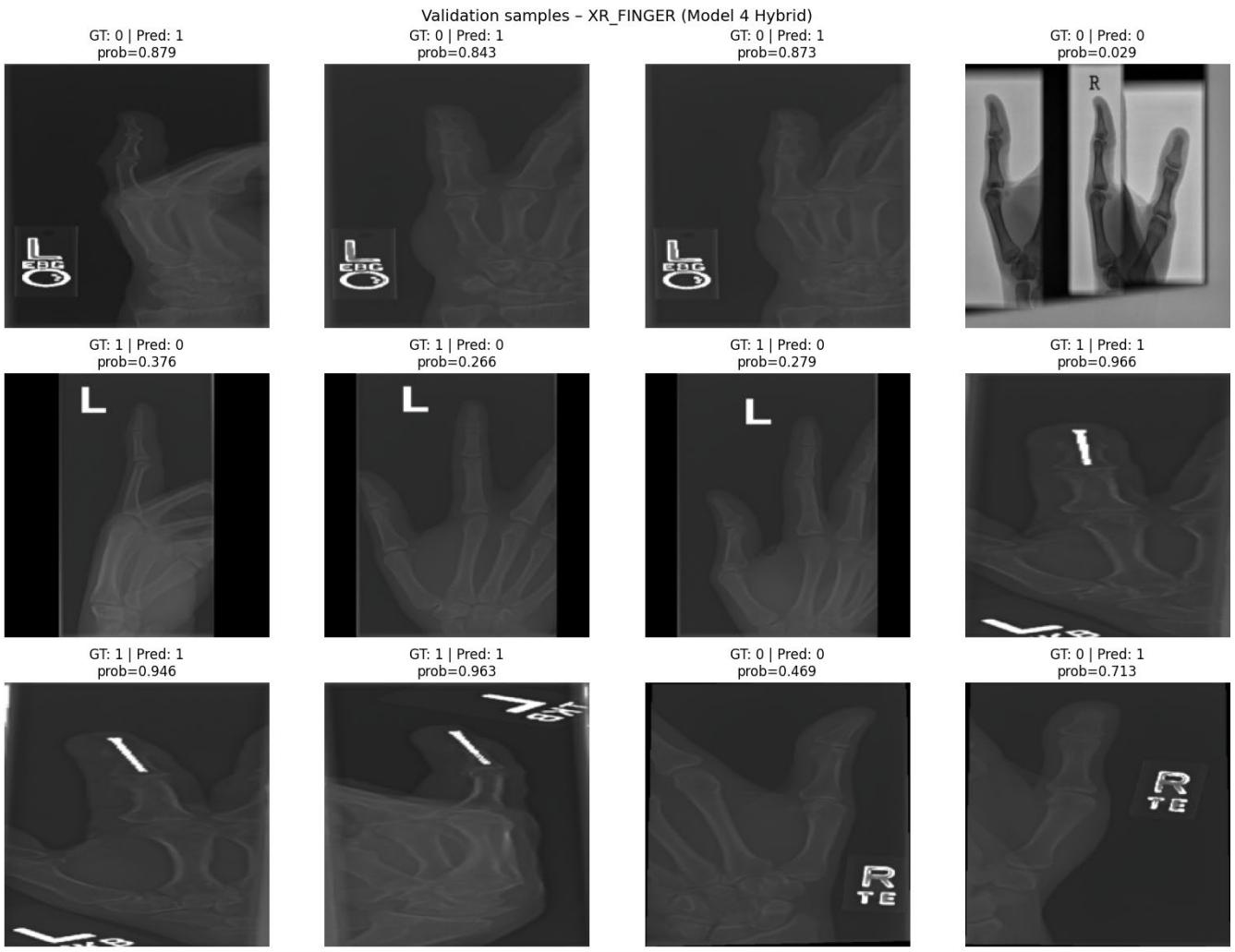
area	XR_HAND
AUC_study	0.8017
ACC_study	0.7186
F1_study	0.5347
val_size_img	460
pos_count	189
neg_count	271

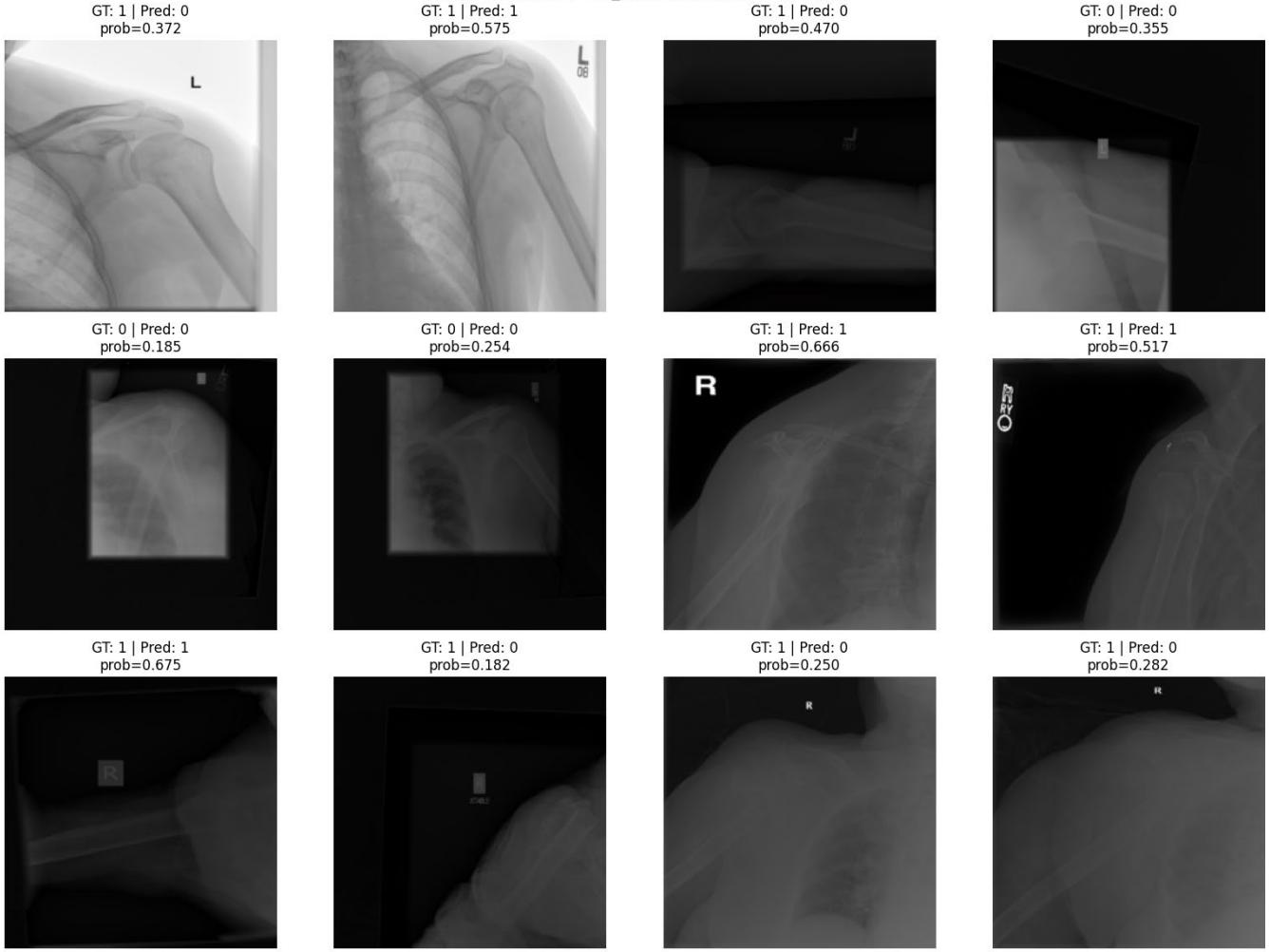
Рисунок 42(в) - BEST AREA (BY STUDY AUC)

Рисунок 42(г) - WORST AREA (BY STUDY AUC)

По агрегированным метрикам модель показывает умеренное качество ранжирования, но слабее по качеству решений при фиксированном пороге: на image-level: AUC macro=0.7909, ACC=0.7186, Precision=0.7515, Recall=0.6056, F1=0.6606; на study-level: AUC macro=0.8111, ACC=0.7363, Precision=0.7689, Recall=0.5964, F1=0.6594. Ключевой перекос - пониженный Recall (≈ 0.60), то есть модель чаще пропускает позитивы, из-за чего F1 не растет.

Лучшая область по study-AUC - XR_SHOULDER (AUC_study=0.8235, ACC_study=0.7474, F1_study=0.7066, val_size=563; pos=278 / neg=285). Это самая стабильная область в рамках гибрида: и ранжирование, и итоговая точность заметно выше, чем у остальных двух. Худшая область - XR_HAND (AUC_study=0.8017, ACC_study=0.7186, F1_study=0.5347, val_size=460; pos=189 / neg=271). Здесь при относительно приемлемом AUC итоговый F1 резко ниже, что типично для ситуации, когда при текущем пороге модель либо недобирает Recall, либо дает несбалансированные ошибки по классам.





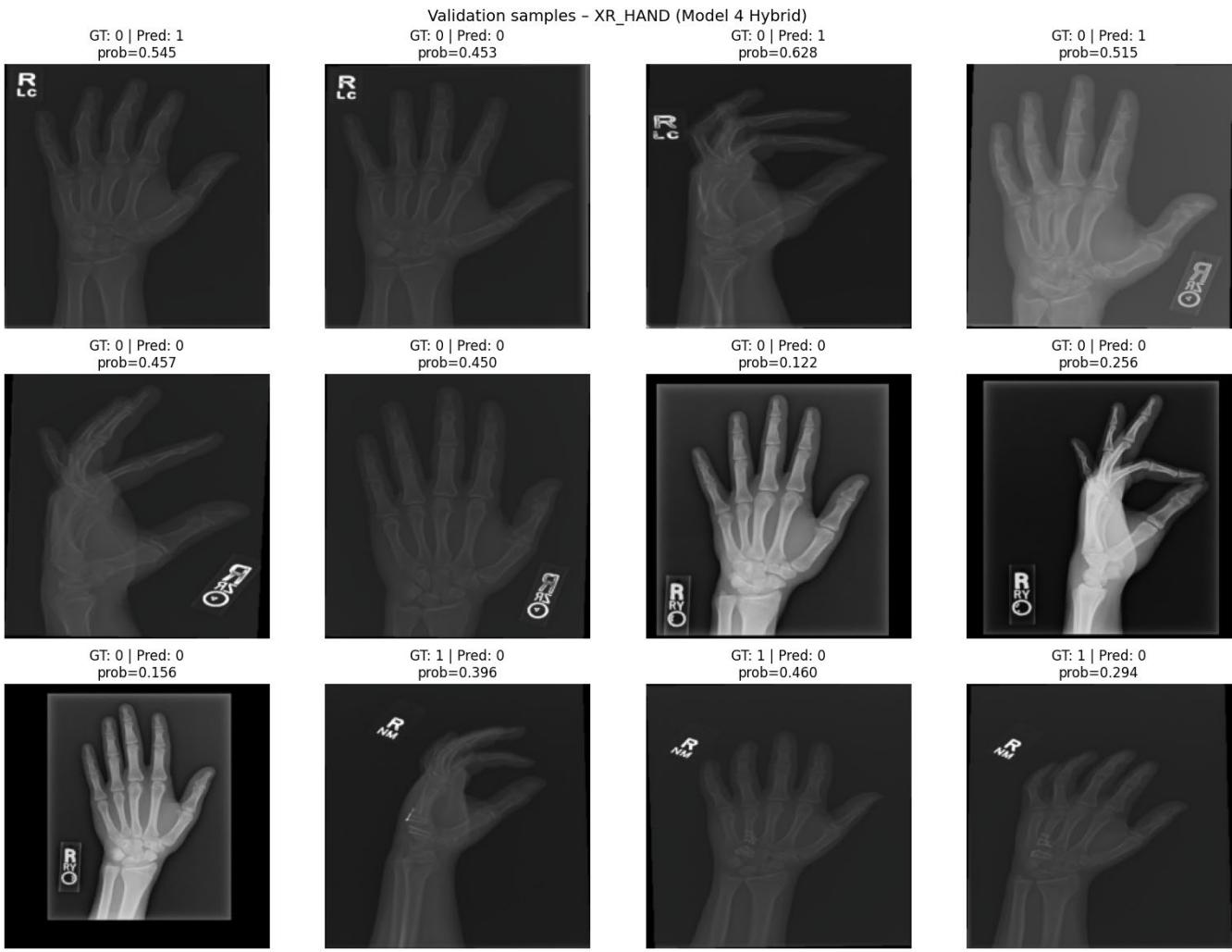


Рисунок 43 - 3 "XR_FINGER", "XR_HAND", "XR_SHOULDER"

По валидационным примерам (GT / Pred / prob) можно сделать вывод о характере ошибок и уровне уверенности модели в каждой из трех областей.

Для XR_FINGER заметна группа ложноположительных случаев (GT=0, Pred=1) с высокой апостериорной вероятностью ($prob \approx 0.84-0.88$). Это означает, что модель не просто ошибается, а принимает решение с высокой уверенностью, интерпретируя некоторые нормальные варианты/артефакты как признаки патологии. Одновременно встречаются ложные отрицания (GT=1, Pred=0) со значениями вероятности ниже порога, но не экстремально низкими ($prob \approx 0.27-0.38$), что соответствует слабому сигналу: модель частично улавливает патологию, но оценка остается недостаточной для классификации как положительного класса. Корректные предсказания представлены как уверенными отрицательными (GT=0, Pred=0; $prob \approx 0.03$), так и уверенными положительными (GT=1, Pred=1; $prob \approx 0.95$).

0.97). В целом по примерам для XR_FINGER ошибки имеют смешанную природу: часть - уверенные FP (возможная переинтерпретация визуальных паттернов), часть - FN в зоне умеренной неопределенности около порога.

Для XR_SHOULDER на примерах доминирует сценарий пропусков положительного класса: встречаются GT=1, Pred=0 при $\text{prob} \approx 0.18-0.47$, то есть модель часто оценивает положительные исследования как «скорее отрицательные» или пограничные и не пересекает порог 0.5. Корректные отрицательные случаи (GT=0, Pred=0) сопровождаются низкими вероятностями ($\text{prob} \approx 0.25-0.36$), что выглядит согласованно с решением. Корректные положительные (GT=1, Pred=1) присутствуют, но их уверенность умеренная ($\text{prob} \approx 0.52-0.68$) и близка к порогу, что указывает на слабую separability классов по данному признаковому представлению: даже верные положительные решения часто принимаются на границе. Так, для плеча основной вклад в ошибки вносит недостаточная уверенность модели по позитивному классу и высокая чувствительность к выбору порога.

Для XR_HAND примеры демонстрируют наиболее выраженную нестабильность решений в окрестности порога. Наблюдаются ложноположительные GT=0, Pred=1 с вероятностями $\text{prob} \approx 0.51-0.63$, то есть отрицательные случаи регулярно переходят порог при небольшой прибавке в score. Параллельно присутствуют ложные отрицания GT=1, Pred=0 с вероятностями $\text{prob} \approx 0.29-0.46$ - положительные случаи нередко остаются ниже порога, хотя модель присваивает им не минимальные значения. Это указывает на недостаточное разделение распределений вероятностей между классами: значительная доля примеров обоих классов получает промежуточные оценки, из-за чего ошибки возникают в обе стороны и итог сильно зависит от выбранного threshold. На уровне наблюдений это согласуется с тем, что для кисти модель чаще сталкивается с вариативностью проекций, мелкими структурными изменениями и неоднозначными визуальными паттернами, приводящими к пограничным оценкам.

Сравнительный анализ моделей

area	AUC_study_best_model	AUC_study_best_value	ACC_study_best_model	ACC_study_best_value	Precision_study_best_model	Precision_study_best_value
XR_ELBOW	Model3_FinalDenseNet	0.9181	Model3_FinalDenseNet	0.8608	Model1_Baseline	
XR_FINGER	Model3_FinalDenseNet	0.8941	Model3_FinalDenseNet	0.8114	Model1_Baseline	
XR_FOREARM	Model3_FinalDenseNet	0.9076	Model3_FinalDenseNet	0.8421	Model1_Baseline	
XR_HAND	Model3_FinalDenseNet	0.8624	Model3_FinalDenseNet	0.7844	Model3_FinalDenseNet	
XR_HUMERUS	Model3_FinalDenseNet	0.9333	Model3_FinalDenseNet	0.8741	Model3_FinalDenseNet	
XR_SHOULDER	Model3_FinalDenseNet	0.8893	Model3_FinalDenseNet	0.8093	Model2_CAM_Mixup	
XR_WRIST	Model3_FinalDenseNet	0.9214	Model3_FinalDenseNet	0.8692	Model2_CAM_Mixup	
Precision_study_best_value	Recall_study_best_model	Recall_study_best_value	F1_study_best_model	F1_study_best_value		
0.9185	Model3_FinalDenseNet		0.8333	Model3_FinalDenseNet		0.8333
0.8446	Model3_FinalDenseNet		0.8193	Model3_FinalDenseNet		0.8047
0.9160	Model3_FinalDenseNet		0.7812	Model3_FinalDenseNet		0.8264
0.8571	Model1_Baseline		0.7778	Model1_Baseline		0.7017
0.8788	Model2_CAM_Mixup		0.9403	Model2_CAM_Mixup		0.8811
0.8354	Model3_FinalDenseNet		0.7789	Model3_FinalDenseNet		0.8000
0.9429	Model3_FinalDenseNet		0.7835	Model3_FinalDenseNet		0.8306

Рисунок 44 - Лучшие результаты по моделям

Сравнение моделей на уровне исследования (study-level) показывает, что в итоговой постановке задачи наилучшее и наиболее устойчивое качество обеспечивает Model3_FinalDenseNet. По ключевым метрикам, отражающим способность модели разделять классы и корректно ранжировать исследования по риску патологии, именно Model3 демонстрирует лидерство: она имеет максимальные значения AUC_study и ACC_study во всех семи анатомических областях, включая наиболее сильные результаты на XR_HUMERUS (AUC_study = 0.9333) и XR_WRIST (AUC_study = 0.9214), а также сохраняет преимущество даже на наиболее сложной области XR_HAND (AUC_study = 0.8624). Тем самым Model3 является оптимальным выбором как финальная модель, если целью является максимизация дискриминативной способности и общей диагностической точности на уровне исследования, что и является наиболее релевантным критерием для практического сценария интерпретации серии снимков в рамках одного исследования.

При этом анализ best model по отдельным метрикам показывает ожидаемые компромиссы между типами ошибок. В части precision_study локальные максимумы в ряде областей достигаются более простыми или более «консервативными» вариантами (например, Model1_Baseline и Model2_CAM_Mixup), что указывает на склонность этих моделей реже выдавать положительный класс без высокой уверенности. Однако данные преимущества по precision не сопровождаются

улучшением AUC_study или ACC_study, то есть отражают скорее смещение рабочей точки (порогового режима) и иной баланс ложноположительных/ложноотрицательных, чем более качественное разделение классов. Аналогично, по F1_study в отдельных областях наблюдаются частные выигрыши альтернативных моделей (например, для XR_HAND лучший F1_study достигается Model1_Baseline, а для XR_HUMERUS - Model2_CAM_Mixup), что согласуется с природой F1 как порогозависимой метрики: при фиксированном пороге модель может оказаться более выгодной по балансу precision/recall, даже если ее ранжирующая способность (AUC) ниже.

Наконец, гибридный подход Model4_Hybrid в текущей реализации не демонстрирует конкурентоспособности относительно финальной DenseNet: по доступным областям он имеет существенно более низкие значения AUC и сводных показателей, что указывает на недостаточную зрелость архитектуры/настроек и необходимость дополнительной доработки (обучение в сопоставимых условиях на всех областях, выравнивание режима агрегации и подбора гиперпараметров), прежде чем рассматривать его как потенциальную замену или улучшение.

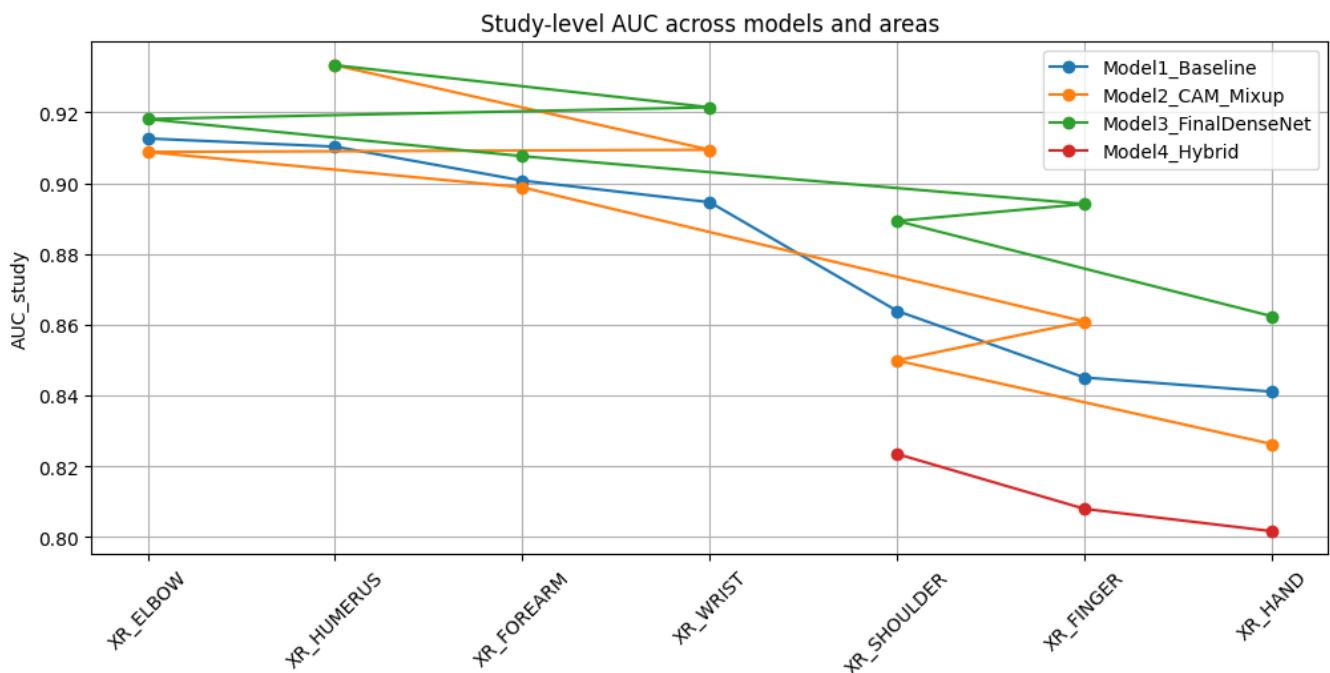


Рисунок 44 – Study-level по моделям

В результате общий вывод по сравнительному анализу таков: Model3_FinalDenseNet является основным финальным решением по критерию качества на уровне исследования (AUC/ACC), тогда как Model1/Model2 могут быть полезны как альтернативы для отдельных областей при специфических требованиях к балансу ошибок (precision или F1), но не изменяют общий порядок моделей по главному показателю диагностической разделимости.

Выводы

В рамках работы решена задача автоматического выявления патологий на рентгенограммах опорно-двигательного аппарата на датасете MURA в постановке per-area с оценкой на уровнях image-level и study-level. Построен полный воспроизводимый пайплайн: проверка структуры данных, предобработка и аугментации, обучение и выбор лучшего чекпойнта по валидации, а также расширенный анализ качества (ROC/PR-кривые, матрицы ошибок, распределения вероятностей и калибровка). В качестве базового решения использована DenseNet; дополнительно исследованы стратегии регуляризации (MixUp/CutMix) и методы интерпретируемости семейства CAM/Grad-CAM, позволяющие качественно оценивать, на какие области снимка опирается модель и где возникают типовые ошибки.

Наиболее сильные результаты показала финальная модель DenseNet с двухфазным дообучением (Model3_FinalDenseNet), обеспечившая лучшее качество на уровне исследования практически по всем анатомическим областям и ключевым метрикам. По aggregated study-level показателям достигнуты значения AUC_study macro = 0.9038 и weighted ≈ 0.9026 при ACC_study macro = 0.8359 и F1_study macro = 0.8049; при этом лучшая область - XR_HUMERUS (AUC_study = 0.9333), а наиболее сложная - XR_HAND (AUC_study = 0.8624). Сравнительный анализ показал, что альтернативные варианты (baseline и CAM+MixUp) могут давать локальные выигрыши по порогозависимым метрикам (precision или F1) в отдельных областях, однако по основному критерию диагностической разделимости на уровне исследования (AUC_study) уступают финальной DenseNet. Гибридная архитектура

(Model4_Hybrid) в текущей реализации не достигла качества Model3 и требует дополнительной настройки, обучения в сопоставимых условиях и более строгого подбора гиперпараметров, прежде чем рассматривать её как улучшение.

Отдельно отмечено, что помимо качества классификации важной практической характеристикой является надежность вероятностных оценок: анализ калибровки выявил заметную рассогласованность предсказанных вероятностей и эмпирической точности в ряде областей, что указывает на необходимость посткалибровки при использовании модели в сценариях, где требуется интерпретируемая уверенность. Интерпретационный анализ на основе CAM подтвердил, что часть ошибок связана с фокусировкой на нерелевантных артефактах (рамки, маркеры, области коллимации, элементы фиксации), а также с неоднозначными случаями и перекрытиями анатомических структур; такие наблюдения задают направления для дальнейшего улучшения данных и обучения.

Заключение и направления дальнейших исследований

В ходе выполнения работы разработан и реализован полный пайплайн обучения и оценки моделей глубокого обучения для автоматического выявления патологий на рентгенограммах опорно-двигательного аппарата (датасет MURA) в постановке *per-area*. Проведена подготовка данных, построены процедуры обучения и валидации, реализовано вычисление метрик на уровнях *image-level* и *study-level*, а также выполнен расширенный пост-анализ результатов с использованием ROC/PR-кривых, матриц ошибок, распределений предсказанных вероятностей и диаграмм надёжности. Дополнительно реализованы средства интерпретируемости на основе методов семейства CAM/Grad-CAM и проведён качественный анализ типовых ошибочных предсказаний.

Сравнительный анализ нескольких конфигураций показал, что наилучшее качество на уровне исследования обеспечивает финальная DenseNet с двухфазным дообучением (Model3_FinalDenseNet). Данная модель демонстрирует стабильное превосходство по AUC_study в большинстве анатомических областей и обеспечивает наилучший баланс метрик при фиксированном пороге, что делает ее

наиболее практичной как базовое решение для дальнейшего развития. Гибридная архитектура (Model4_Hybrid) в текущем виде уступает лучшей DenseNet по интегральным метрикам, что указывает на необходимость дополнительной настройки и более тщательного подбора режимов обучения, прежде чем рассматривать ее как улучшение. Анализ калибровки показал, что вероятностные оценки модели не всегда отражают истинную уверенность, а интерпретационный анализ выявил вклад нерелевантных визуальных факторов и артефактов в часть ошибок, что задаёт направления для последующей оптимизации.

Направления дальнейших исследований

Дальнейшая работа может быть направлена на повышение качества, устойчивости и практической применимости решения. Ниже приведены наиболее перспективные и конкретные направления.

1) Доведение гибридного подхода до сопоставимых условий и «честного» сравнения. Текущая гибридная модель оценивалась только на трех областях и имеет заметный зазор относительно Model3, поэтому первым шагом является приведение экспериментов к единому протоколу:

- обучить Model4 для всех областей на одинаковых разбиениях и числе эпох, с теми же аугментациями и стратегией early stopping/лучшего чекпойнта;
- провести систематический подбор гиперпараметров именно для гибрида (веса регуляризации, learning rate, scheduler, dropout, балансировка классов), а не переносить настройки DenseNet из коробки;
- оценивать не только AUC, но и F1/Recall в клинически важной области порогов (например, фиксируя Recall или ограничивая FPR), чтобы понять, где гибрид реально полезен.

2) Переход от «простого объединения признаков» к более выразительной схеме слияния Если гибрид строится как конкатенация признаков DenseNet + Swin

(или другого трансформера), то модель может не использовать дополнительные признаки эффективно. Улучшения:

- заменить простую конкатенацию на gated fusion / attention fusion, где сеть учится динамически взвешивать вклад CNN и трансформера по каждому примеру;
- использовать multi-scale fusion (слияние на нескольких уровнях пирамиды признаков), а не один вектор в конце;
- добавить distillation: обучать трансформерный блок как «ученика», повторяющего сильную DenseNet (или наоборот), что часто стабилизирует обучение на ограниченных данных.

3) Улучшение качества именно в «трудных» областях (XR_HAND, XR_SHOULDER, XR_FINGER) через анализ ошибок и таргетированные меры:

- сформировать набор «hard cases» (FN и FP с высокой уверенностью) и отдельно проанализировать типовые причины (качество снимка, нестандартная укладка, металлические конструкции, рамки/маркировка);
- усилить аугментации не «в целом», а адресно: имитация рамок, надписей, различной коллимации, измененного контраста/шума, motion blur;
- применить hard example mining или веса примеров по сложности, чтобы модель чаще видела проблемные случаи.

4) Моделирование на уровне исследования как отдельная задача, а не «сумма картинок». Study-level качество является ключевым, но оно чувствительно к тому, как агрегируются предсказания по изображениям исследования. Улучшения:

- заменить простую агрегацию на learned pooling: attention pooling по изображениям исследования;
- использовать MIL (multiple instance learning): считать исследование мешком изображений и обучать модель напрямую на study-level метке;

- анализировать устойчивость к количеству/качеству снимков в исследовании, если часть изображений низкого качества.

5) Калибровка вероятностей и критерии принятия решения. Даже при высоком AUC модель плохо калибрована, а для практического использования требуется интерпретируемая уверенность:

- применить temperature scaling и сравнить ECE до/после;
- выбирать пороги не фиксированно 0.5, а по целевому режиму: максимум F1, фиксированный Recall (минимизация FN), или ограничение FPR;
- дополнить оценками доверия: Brier score, калибровочные кривые, разбор случаев «высокая уверенность, но ошибка».

6) Повышение устойчивости к «shortcut-признакам» и артефактам разметки. CAM-анализ показывает, что модель может опираться на маркеры, рамки, области фона. Возможные меры:

- регуляризация внимания: случайное подавление наиболее активных областей в строго контролируемом режиме и сравнение с базой;
- маскирование маркеров/рамок или их синтетическое добавление в обеих классах;
- обучение с domain randomization: вариативность яркости/ контраста/ фоновых артефактов, чтобы сеть не «привязывалась» к частным признакам.

7) Архитектурные улучшения без полной смены модели Если цель - улучшить Model3, не уходя в трансформеры полностью:

- более сильные backbone с тем же протоколом;
- self-supervised pretraining на MURA-подобных данных и последующий fine-tune;
- ансамблирование.

Список литературы:

1. Rajpurkar, P., Irvin, J., Bagul, A., Ding, D., Duan, T., Mehta, H., Yang, B., Zhu, K., Laird, D., Ball, R. L., Langlotz, C., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2018). MURA: Large dataset for abnormality detection in musculoskeletal radiographs. *Proceedings of the 1st Conference on Medical Imaging with Deep Learning (MIDL 2018)*. // arXiv. 2017. DOI: 10.48550/arXiv.1712.06957. URL: <https://arxiv.org/abs/1712.06957> (дата обращения: 16.11.2025). PDF: <https://arxiv.org/pdf/1712.06957.pdf>.
2. Deng, Jia, Dong, Wei, Socher, Richard, Li, Li-Jia, Li, Kai, and Fei-Fei, Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. IEEE, 2009.
3. PyTorch. Torchvision documentation: densenet121 (pretrained weights and recommended inference transforms) [Электронный ресурс]. URL: <https://docs.pytorch.org/vision/stable/models/generated/torchvision.models.densenet121.html> (дата обращения: 17.11.2025)
4. Huang G., Liu Z., van der Maaten L., Weinberger K. Q. Densely Connected Convolutional Networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017. P. 2261–2269. DOI: 10.1109/CVPR.2017.243. URL: https://openaccess.thecvf.com/content_cvpr_2017/papers/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.pdf (дата обращения: 17.11.2025)
5. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework // Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019.
6. https://colab.research.google.com/drive/1_-WVmLMjwgWtlTaQt6lZZk_Qdb7-Mfp4?usp=sharing – открытый исходный код работы.