

метрики в computer vision

Отличный вопрос! Понимание метрик критически важно для соревнований. Вот подробный разбор основных метрик, сгруппированных по задачам.

Классификация изображений

1. Accuracy (Точность)

- **Формула:** $(TP + TN) / (TP + TN + FP + FN)$
- **Что измеряет:** Долю правильных предсказаний среди всех.
- **Плюсы:** Интуитивная, простая.
- **Минусы: Бесполезна при дисбалансе классов.** Если 99% данных — класс "A", а модель всегда предсказывает "A", accuracy = 0.99, но модель бесполезна.
- **Когда использовать:** Только если классы сбалансированы и ошибки всех типов одинаково важны.

2. Precision (Точность/Прецизионность)

- **Формула:** $TP / (TP + FP)$
- **Что измеряет:** Из всех объектов, которые модель назвала положительными (класс X), какая доля действительно положительна. **"Качество положительных предсказаний".**
- **Когда важна:** Когда **ложноположительные (FP)** дорого обходятся.
 - **Пример:** Спам-фильтр. Ложно пометить важное письмо как спам (FP) — плохо.

3. Recall (Полнота)

- **Формула:** $TP / (TP + FN)$
- **Что измеряет:** Из всех реальных положительных объектов, какую долю модель нашла. **"Способность находить все нужное".**
- **Когда важна:** Когда **ложноотрицательные (FN)** критичны.
 - **Пример:** Диагностика рака. Пропустить больного (FN) — катастрофа.

4. F1-Score (F-мера)

- **Формула:** $2 * (Precision * Recall) / (Precision + Recall)$
- **Что измеряет:** Гармоническое среднее Precision и Recall. Баланс между ними.
- **Плюсы:** Удобная единая метрика при дисбалансе классов.

- **Минусы:** Не учитывает True Negatives. Может маскировать плохие значения одной из метрик, если другая очень высока.
- **Варианты:** F β -Score, где β определяет важность Recall относительно Precision ($\beta > 1$ — Recall важнее).

5. ROC-AUC (Area Under the ROC Curve)

- **ROC-кривая (Receiver Operating Characteristic):** График зависимости **True Positive Rate (TPR = Recall)** от **False Positive Rate (FPR = FP / (FP + TN))** при изменении порога классификации.
- **AUC (Area Under Curve):** Площадь под ROC-кривой. **Вероятность того, что случайно выбранный positive-пример будет иметь оценку (score) выше, чем случайно выбранный negative-пример.**
- **Плюсы:** Не зависит от порога классификации и баланса классов. Показывает качество ранжирования.
- **Минусы:** Может быть оптимистичной при сильном дисбалансе (т.к. FPR на маленьких отрицательных выборках менее информативен).
- **Когда использовать:** Для бинарной классификации, когда важна **сепаративность** модели. Одна из самых популярных метрик.

6. PR-AUC (Area Under the Precision-Recall Curve)

- **PR-кривая:** График зависимости Precision от Recall при изменении порога.
- **AUC-PR:** Площадь под PR-кривой.
- **Плюсы: Лучше, чем ROC-AUC, при сильном дисбалансе классов.**
Концентрируется на positive-классе.
- **Когда использовать:** Почти всегда при дисбалансе (например, 1% positive, 99% negative).

7. Log Loss (Binary Cross-Entropy Loss)

- **Формула:** $-(1/N) * \sum [y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)]$
- **Что измеряет:** Штрафует за **уверенность в неверном предсказании**. Чем дальше предсказанная вероятность от истинного значения, тем больше лосс.
- **Плюсы:** Чувствительная метрика, оценивает "качество" вероятностей.
- **Минусы:** Сильно штрафует за выбросы (очень уверенные ошибки).

8. Multi-class метрики

- **Macro-average:** Вычисляется метрика (например, Precision) для каждого класса независимо, затем усредняется. **Все классы равнозначны.**

- **Micro-average:** Глобально подсчитываются TP, FP, FN по всем классам и вычисляется одна общая метрика. **Взвешивает классы по количеству примеров.**
 - **Weighted-average:** Как Macro, но усреднение с весом, равным поддержке (support) класса. Учитывает дисбаланс.
-

Детекция объектов (Object Detection)

1. IoU (Intersection over Union / Jaccard Index)

- **Формула:** Area of Overlap / Area of Union
- **Что измеряет:** Пересечение предсказанного и истинного bounding box'ов. Значение от 0 (нет пересечения) до 1 (полное совпадение).
- **Применение:** Определяет, считать ли предсказание **True Positive**. Обычно используется порог (threshold), например, $\text{IoU} > 0.5$.

2. mAP (mean Average Precision) — КОРОЛЕВА МЕТРИК ДЛЯ ДЕТЕКЦИИ

- **Основная идея:** Усредненная точность (Average Precision) по всем классам и порогам IoU.
- **Как вычисляется (по шагам):**
 1. Для **одного класса и одного порога IoU** (например, 0.5):
 2. Для каждого изображения собираем все предсказанные bounding boxes с confidence score.
 3. Сортируем все предсказания по confidence (уверенности) **по убыванию**.
 4. Вычисляем **Precision** и **Recall** по мере движения по этому списку, добавляя предсказания одно за другим (считая TP/FP по текущему порогу IoU).
 5. Строим **PR-кривую** (Precision-Recall curve) для этого класса и порога IoU.
 6. Вычисляем **Average Precision (AP)** для этого класса и порога — это площадь под сглаженной PR-кривой.
 7. Для получения **mAP@0.5** усредняем AP по всем классам.
- **Варианты:**
 - **mAP@0.5** (или **mAP50**): Только для IoU порога 0.5.
 - **mAP@0.5:0.95** (или просто **mAP**): Вычисляют AP для **нескольких порогов IoU** (от 0.5 до 0.95 с шагом 0.05), затем усредняют AP по порогам, а потом по классам. **Самая строгая и популярная метрика в соревнованиях (COCO).**
- **Плюсы:** Учитывает и точность локализации (через IoU), и точность классификации, и уверенность модели.

- **Минусы:** Сложна для расчета и интерпретации.

3. Другие метрики для детекции

- **AR (Average Recall):** Часто используется в связке с AP. Средний Recall при фиксированном количестве детекций на изображение (AR@1, AR@10, AR@100) или при разных порогах IoU.
 - **FPS (Frames Per Second):** Метрика скорости, критична для real-time систем.
-

Сегментация (Semantic & Instance Segmentation)

1. Pixel Accuracy

- **Формула:** (Количество правильно классифицированных пикселей) / (Всего пикселей)
- **Аналогия:** Accuracy для классификации, но на уровне пикселей.
- **Недостатки:** Сильно искажается при дисбалансе (например, если 90% изображения — фон).

2. IoU (Intersection over Union) / Jaccard Index — ОСНОВНАЯ МЕТРИКА

- Для семантической сегментации: IoU считается для каждого класса отдельно, между предсказанной маской класса и истинной маской.
 - **Формула (на класс):** $\text{Area of Overlap} / \text{Area of Union} = \text{TP} / (\text{TP} + \text{FP} + \text{FN})$
- Для instance сегментации: Сначала проводят matching между предсказанными и истинными масками (обычно по порогу IoU), а затем вычисляют IoU для matched пар.

3. mIoU (mean IoU)

- Самый популярный бенчмарк для семантической сегментации.
- Вычисление: 1) Вычислить IoU для каждого класса. 2) Усреднить по всем классам.
- Плюсы: Интуитивная, устойчива к дисбалансу (т.к. усредняется по классам, а не по пикселям).

4. Dice Coefficient (F1-Score для пикселей)

- **Формула:** $(2 * |A \cap B|) / (|A| + |B|) = 2 * \text{TP} / (2 * \text{TP} + \text{FP} + \text{FN})$

- **Связь с IoU:** $\text{Dice} = (2 * \text{IoU}) / (1 + \text{IoU})$
- **Особенность:** Сильнее штрафует за несовпадение, чем IoU. Широко используется в медицинской сегментации (биомедицинские изображения).

5. PQ (Panoptic Quality) — для Panoptic Segmentation

- **Panoptic Segmentation** = Semantic + Instance Segmentation (каждому пикслю присваивается и класс, и ID экземпляра).
 - **Формула:** $\text{PQ} = (\Sigma \text{ по классам } (\text{PQ}_c)) / (\text{Количество классов})$, где $\text{PQ}_c = (\Sigma \text{ по совпадшим парам IoU}) / (\text{Количество объектов в GT} + 0.5*\text{FP} + 0.5*\text{FN})$
 - **Интерпретация:** Учитывает и качество сегментации (IoU), и качество обнаружения экземпляров.
-

Регрессия (например, определение ключевых точек, восстановление глубины)

1. MSE (Mean Squared Error) / L2 Loss

- **Формула:** $(1/N) * \sum (y_i - \hat{y}_i)^2$
- **Что измеряет:** Средний квадрат ошибок.
- **Особенность:** Сильно штрафует за большие ошибки (из-за квадрата). Может быть чувствительна к выбросам.

2. RMSE (Root Mean Squared Error)

- **Формула:** $\text{sqrt}(\text{MSE})$
- **Преимущество:** Имеет ту же размерность, что и целевая переменная (например, пиксели), что делает интерпретацию проще.

3. MAE (Mean Absolute Error) / L1 Loss

- **Формула:** $(1/N) * \sum |y_i - \hat{y}_i|$
- **Что измеряет:** Среднюю абсолютную ошибку.
- **Сравнение с MSE:** Менее чувствительна к выбросам, чем MSE. Штрафует линейно.

4. MAPE (Mean Absolute Percentage Error)

- **Формула:** $(100\% / N) * \sum |(y_i - \hat{y}_i) / y_i|$
- **Что измеряет:** Среднюю абсолютную ошибку в процентах.

- **Недостаток:** Не определен, когда истинное значение $y_i = 0$. Может быть нестабильной при малых значениях.

5. PCK (Percentage of Correct Keypoints)

- **Для pose estimation:** Ключевая точка считается корректной, если она находится в пределах заданного расстояния (часто пропорционального размеру человека) от истинной точки.
 - **Пример:** [PCK@0.2](#) — процент ключевых точек, ошибка которых $\leq 0.2 * \text{референсного размера}$.
-

Практические советы для соревнований

1. **ПОНИМАЙТЕ МЕТРИКУ СОРЕВНОВАНИЯ В ПЕРВУЮ ОЧЕРЕДЬ!** Ваша стратегия должна быть направлена на ее оптимизацию.
2. **Всегда создавайте локальную валидацию**, которая надежно оценивает метрику (через кросс-валидацию).
3. **Контролируйте overfitting на Public Leaderboard**. Часто метрика на приватном teste (private LB) оказывается хуже.
4. **Для выбора модели** используйте валидационную метрику, но также смотрите на **качественные результаты** (визуализируйте ошибки).
5. **Распространенные пары "задача-метрика":**
 - Классификация (баланс) → **Accuracy / F1**
 - Классификация (дисбаланс) → **ROC-AUC / PR-AUC / F1**
 - Детекция объектов → [mAP@0.5:0.95 \(COCO style\)](#)
 - Семантическая сегментация → **mIoU**
 - Instance сегментация → **mAP (по маскам)**
 - Медицинская сегментация → **Dice Coefficient**
 - Регрессия (ключевые точки) → **PCK / MSE**

Для Kaggle: В описании каждого соревнования есть раздел "Evaluation". Изучите его, найдите формулы, посмотрите готовый код для расчета метрики — это ваш ключ к успеху.