

МОДЕЛЬ РАСПОЗНАВАНИЯ ПСИХОЭМОЦИОНАЛЬНОГО СОСТОЯНИЯ ЧЕЛОВЕКА В РЕАЛЬНОМ ВРЕМЕНИ

Е.М. Косарева

Научный руководитель Д.В. Лихачевский

*Белорусский государственный университет информатики
и радиоэлектроники, Минск, Беларусь, kksrvvv@gmail.com*

При разработке систем интеллектуального видеонаблюдения как компонента системы распознавания потенциальных угроз в рамках «умного города» особое внимание уделяется предиктивной аналитике антропогенных угроз. Согласно [1, с. 168], распознавание антропогенных угроз может производиться на основании оценки текущего психоэмоционального состояния лица, потенциально реализующего социальную опасность.

В настоящем исследовании используется модель для распознавания эмоций на основе библиотеки Facial Emotion Recognition

(FER), реализация которой выполнена с использованием Keras. FER представляет собой пакет с открытым исходным кодом, предназначенный для классификации эмоций по изображениям человеческих лиц. Под «эмоциями» здесь подразумевается набор из семи базовых состояний, обычно включающих *angry*, *disgust*, *fear*, *happy*, *sad*, *surprise* и *neutral*. Сетевые веса модели по умолчанию хранятся в формате HDF5, что характерно для Keras-проектов на базе сверточных нейронных сетей (CNN).

Внутренняя архитектура модели FER, как правило, включает несколько сверточных слоёв (*convolutional layers*), за которыми следуют пулинг-слои (*pooling layers*), ускоряющие вычисления и предотвращающие переобучение. После нескольких таких этапов извлечения признаков идет полносвязная часть (*fully connected layers*), завершающаяся выходным слоем, отражающим вероятности принадлежности к конкретной эмоции. Модель обучалась на крупном датасете эмоциональных портретов (FER-2013) и, следовательно, способна достаточно надёжно детектировать типичные мимические проявления. В настоящем исследовании эта модель была использована по схеме частичного дообучения (*fine-tuning*) на дополнительной выборке, чтобы повысить точность распознавания в специфических условиях освещения и ракурса.

Логика работы системы для распознавания эмоций строится на последовательной обработке входного видеопотока. Сначала видеопоток захватывается с камеры (применяя модуль *camera.py*, где паттерн *Singleton* гарантирует, что камера создаётся и управляется единственным экземпляром объекта). Далее каждое изображение кадра передаётся на детектор лиц, основанный на *dlib*: библиотека *dlib* предоставляет удобный метод *get_frontal_face_detector()*, который находит координаты лиц в кадре.

Для каждого обнаруженного лица формируется вырезанный фрагмент (ROI — *Region Of Interest*) и передаётся в модель FER. Модель, используя заранее обученные свёрточные фильтры, анализирует изображение, извлекает высокоуровневые признаки (изгибы губ, положение глаз, нахмуренность бровей и т. д.) и на выходе выдаёт вектор вероятностей для каждого класса эмоций. Затем выбирается эмоция с наибольшим значением вероятности в качестве

итоговой. Дополнительно, чтобы избежать скачкообразных изменений, применяется сглаживание (post-processing).

Помимо определения эмоции, система параллельно решает задачу идентификации лиц. Для этого используется механизм генерации дескрипторов лиц (face descriptors), который также предоставляет dlib. Сформированное векторное представление (128-мерный вектор) сравнивается с заранее вычисленными дескрипторами эталонных лиц, хранящихся в локальной базе. Если евклидово расстояние между векторами не превышает заданный порог (например, 0.6), система считает, что лицо уже присутствует в базе, и выводит соответствующую информацию. Так достигается одновременная идентификация человека и определение его эмоционального состояния.

В исходном виде FER поставляется с предобученными весами, полученными на крупных публичных датасетах (включающих десятки тысяч изображений людей с выраженными эмоциями). Однако для уточнения модели под конкретные условия в настоящем исследовании была сформирована дополнительная выборка, включающая около 300 фотографий с различными эмоциями. Эти изображения были либо сняты в условиях офисного освещения, либо взяты из публичных источников с целью повысить вариативность (разные ракурсы, возрастные группы, цветовая гамма и т. д.).

Чтобы провести fine-tuning выборка была разбита на три части:

- обучающая (training) выборка — около 70 % изображений;
- валидационная (validation) выборка — 20 % изображений;
- тестовая (test) выборка — оставшиеся 10 %.

При этом для валидации и теста особое внимание уделялось тому, чтобы в набор попали как «простые» фото (лицо анфас, нормальное освещение), так и более «сложные» (разные наклоны головы, мимика, аксессуары вроде очков). Тем самым обеспечивается большая обобщающая способность модели.

Дообучение модели производилось согласно следующему алгоритму:

- загрузка предобученной модели (FER);
- заморозка части слоёв (base_layers);
- разморозка последних слоев и обучение на своих данных с постепенной корректировкой весов;

- указание оптимизатора (например, Adam) и функции потерь (categorical_crossentropy);
- разбиение данных на обучающую и валидационную выборки, обучение в течение 10 эпох с использованием EarlyStopping, чтобы не допустить переобучения;
- сохранение итоговой модели.

На практике выбор количества эпох и гиперпараметров зависит от объёма данных и аппаратных возможностей. В настоящем исследовании, при использовании около 300–500 собственных изображений, достаточно 5–10 эпох, чтобы донастроить высшие слои нейронной сети.

Для дообучения модели был применен стандартный подход «fine-tuning» в Keras. Ниже приведен упрощенный псевдокод:

```
# Загрузка предобученной модели FER
model = load_pretrained_FER_model(weights='pretrained_
emotion_model.h5')

# Заморозка части сверточных слоёв, чтобы не переобучить
базовые фильтры
for layer in model.base_layers:
    layer.trainable = False

# Разморозка верхних слоев
for layer in model.top_layers:
    layer.trainable = True

# Компиляция модели
model.compile(optimizer=Adam(lr=0.0001),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Загрузка собственного датасета
train_data, val_data = load_data("my_emotion_dataset/", split=(0.7,
0.2))

test_data = load_data("my_emotion_dataset/", split=(0.1),
mode="test")

# Обучение в несколько эпох
model.fit(train_data,
          validation_data=val_data,
          epochs=10,
```

```
batch_size=32,  
        callbacks=[EarlyStopping(patience=3, restore_best_  
weights=True)])
```

```
# Сохранение новой модели
```

```
model.save("my_finetuned_emotion_model.h5")
```

Таким образом, описанный метод позволяет получить эффективную модель на основе Keras, при помощи которой можно производить оценку текущего психоэмоционального состояния человек в реальном времени с высокой степенью достоверности.

Список источников

1. Косарева Е. М., Лихачевский Д. В. Признаки потенциально опасных лиц как одной из угроз в системах «умного» города // Современные проблемы радиоэлектроники и телекоммуникаций. 2024. № 7. С. 168.