Project Report - EE656

Understanding and Implementation of DICE-FER for Identity-Invariant Facial Expression Recognition

---

1. Objective

The primary objective of this project was to gain an in-depth understanding of the methodology proposed in the paper *"Decoupling Identity Confounders for Enhanced Facial Expression Recognition"* and to implement its core components through code. The paper introduces **DICE-FER**, a novel deep learning framework designed to decouple facial expression features from identity-specific confounding factors. This is achieved using **mutual information (MI) estimation**, eliminating the need for identity labels or computationally intensive synthetic image generation. Through this project, we aimed to replicate and apply this identity-invariant expression recognition approach in a practical setting.

---

2. Summary of the Paper

2.1 Problem Statement

Facial Expression Recognition (FER) is often hindered by identity-specific features like face shape or skin tone, which confound expression recognition. Existing methods using GANs, reconstruction, or identity labels are either resource-heavy or impractical.

2.2 Solution: DICE-FER

DICE-FER addresses this by:

- Using mutual information (MI) to separate expression features from identity.
- Training on paired images with the same expression to isolate shared (expression) and exclusive (identity) representations.
- Eliminating the need for identity labels or image generation.

---

3. Technical Concepts

3.1 Mutual Information (MI)

Mutual Information (MI) quantifies how much information one variable contains about another. In DICE-FER, MI is estimated using the **Donsker-Varadhan (DV) representation**:

$$I(M, Z) \approx Ep[M, Z][\ U\theta(m, z)] \ - \ log(Ep[M]p[Z][exp(U\theta(m, z))])$$

Here, UθU_\thetaUθ is a neural network called the **statistics network**, which learns to estimate MI between inputs and their representations.

3.2 Core Components

Core Components

- **Expression Encoder (E_exp):** Learns identity-invariant expression features.
- **Identity Encoder (E_id):** Captures features unique to individual identity.
- **Statistics Networks:** Estimate MI to encourage or discourage shared information.
- **Discriminator (D):** Adversarially minimizes MI between expression and identity representations to enforce disentanglement.

---

4. Implementation Strategy

4.1 Expression Representation Learning

- Paired images MMM and NNN, sharing the same expression, are passed through expression encoders.
- Mutual Information (MI) is **maximized** between image MMM and expression representation of NNN, and vice versa.
- A **swapping trick** exchanges expression features across the pair to suppress identity information.
- The final loss combines MI maximization with L1 distance minimization to align the expression features:

  Final loss: $Lexp = LMIcross - \delta * ||EM - EN||1$

4.2 Identity Representation Learning

- The expression and identity features are **concatenated** to form the total representation.
- MI is **maximized** between the input image and its full representation [E,I][E, I][E,I] to retain relevant details.
- An **adversarial loss** is used to **minimize** MI between expression and identity features, ensuring disentanglement:

$$Ladv = Efake[log(D(E,I))] + Ereal[log(1 - D(E,I))]$$

---

5. Code Modules

The implementation was structured into the following modules:

5.1 Dataset and Preprocessing

- *Custom Dataset Creation (≈1800 images):*

  Due to the inaccessibility of standard FER datasets like CK+, Oulu-CASIA, RAF-DB, and AffectNet (as used in the original DICE-FER paper), and the incompatibility of available low-resolution open-source datasets (such as 48×48-pixel Kaggle datasets) with our model architecture, we created our own custom dataset consisting of approximately **1600** images. To collect this data, we developed an image acquisition script using OpenCV that enables **real-time photo capture** through a webcam.

  Methodology:

  The script initializes the webcam and continuously reads frames. Each frame is cropped to a 250×250 pixel region to ensure a focused view of the face. A keypress-based mapping system was implemented to allow users to label the expression being captured. Pressing specific keys (e.g., 'h' for happy, 's' for sad) stores the current frame into a corresponding subfolder under the defined dataset directory (e.g., `Photos/Happy`, `Photos/Sad`). Each saved image is uniquely named using a UUID to avoid filename conflicts. The dataset is categorized into seven expression classes: **Neutral, Angry, Happy, Disgust, Sad, Surprise, and Fear**. This interactive and systematic data collection approach allowed us to create a

```python
# Map keypresses to emotion labels
key_map = {
    'n': 'Neutral',
    'a': 'Angry',
    'h': 'Happy',
    'd': 'Disgust',
    's': 'Sad',
    'p': 'Surprise',
    'f': 'Fear'
}

# Start webcam
cap = cv2.VideoCapture(0)
```

controlled, expression-labeled dataset suitable for training the DICE-FER model.

```
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("Failed to grab frame.")
        break

    # Ensure the frame is large enough
    if frame.shape[0] < 370 or frame.shape[1] < 450:
        print("Frame size too small for cropping.")
        continue

    # Crop frame to 250x250 pixels
    frame = frame[120:370, 200:450]

    # Display the cropped frame
    cv2.imshow('Image Collection', frame)

    # Read keypress once
    key = cv2.waitKey(1) & 0xFF

    if key == ord('q'):
        break

    elif chr(key) in key_map:
        emotion = key_map[chr(key)]
        imgname = os.path.join(base_path, emotion, f'{uuid.uuid1()}.jpg')
        cv2.imwrite(imgname, frame)
        print(f'[INFO] Saved: {imgname}')
```

- *MTCNN Preprocessing Pipeline:*

1. Used **MTCNN** (from `facenet_pytorch`) to detect, crop, and align faces.
2. Input images were organized in subfolders by expression labels. All images resized to a uniform **112×112** resolution.
3. Applied quality checks for faces below a **minimum size threshold** and faces with **distorted aspect ratios**.
4. High-quality faces were converted to grayscale and saved into structured directories for training.

5.2 Network Architectures

- *Encoders:*
  Based on ResNet-18, used for both expression and identity feature extraction.

- *Statistics Networks:*
  Two-layer fully connected (FC) networks used to estimate mutual information.

- *Discriminator:*
  Three-layer FC network used to adversarially minimize MI between expression and identity features.

- *Classifier:*
  FC layers trained on disentangled expression vectors for final expression classification.

5.3 Training Pipeline

Stage 1:

- Train **expression encoder (E_exp)** and **statistics networks (U)**.
- Objective: Maximize mutual information (MI) between paired images' expressions.
- Add **L1 loss** to enforce similarity between expression features.

Stage 2:

- Freeze **E_exp**, and train **identity encoder (E_id)**, **statistics networks (U)**, and **discriminator (D)**.
- Objective:
  - Maximize MI between image and total features [E,I][E, I][E,I].
  - **Minimize MI** between expression and identity via adversarial loss.

Final Step:

- Train a **classifier** on the learned expression features for FER.

---

6. Experimental Outcomes

6.1 Evaluation Metric – Modified Mutual Information Gap (MIG)

- *MIG Formula:*

  $MIG = I(EM, EN) - I(EM, IM)$, where:

  I(EM,EN): MI between expression features of paired images

  I(EM,IM)): MI between expression and identity features

  Interpretation:

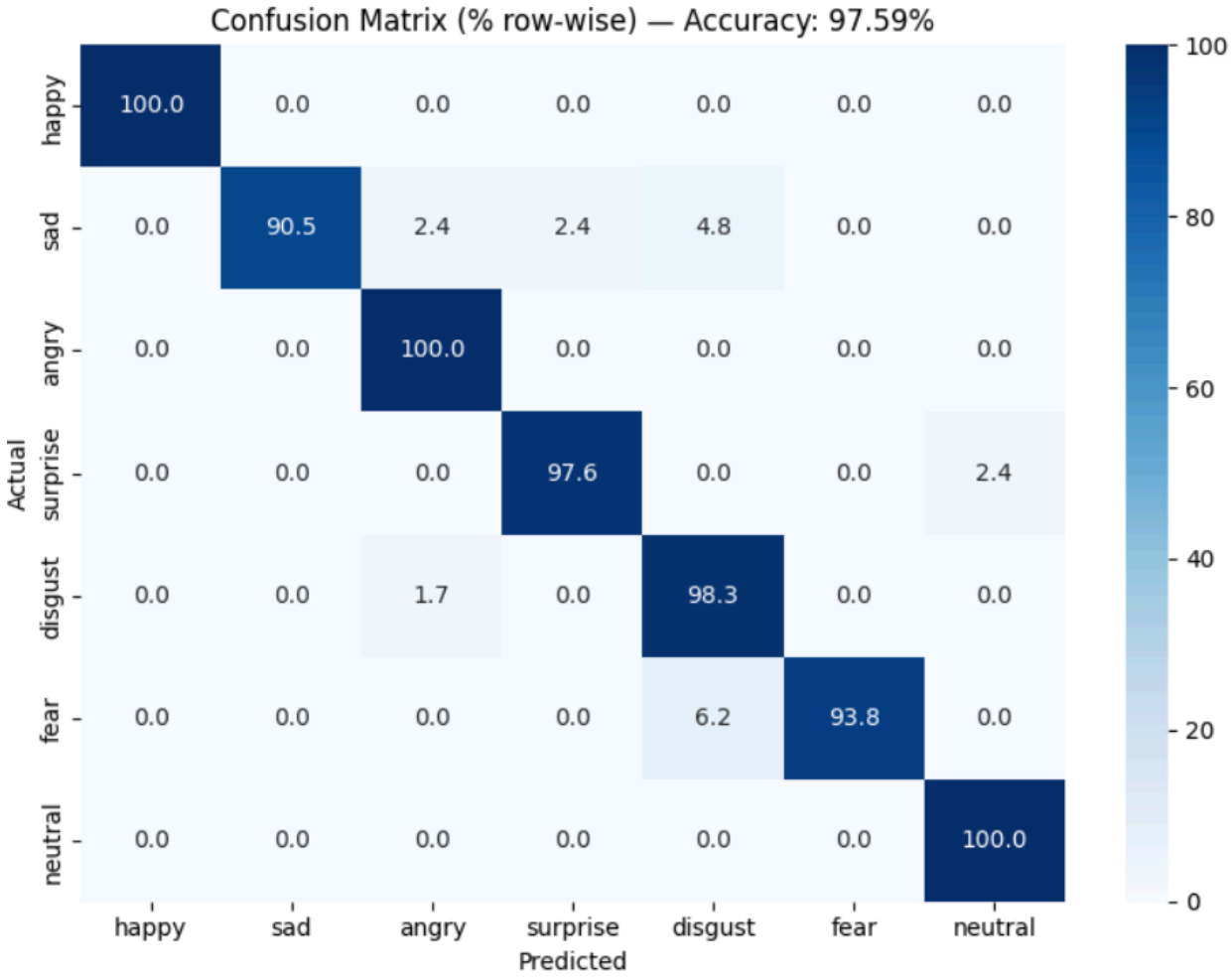
  Higher MIG indicates stronger disentanglement of identity from expression.

6.2 Results

We evaluated our implementation of the DICE framework on a manually curated facial expression dataset containing seven expression classes. The model was trained in three stages: expression encoding, identity disentanglement, and final classification. After training, the expression classifier achieved an accuracy of over **~99.5%** on the train set while achieving an accuracy of **~97%** on the test set, demonstrating the effectiveness of the staged disentanglement process.

To further evaluate disentanglement quality, we computed the **Mutual Information Gap (MIG)**. Expression embeddings showed a significantly higher MIG score compared to identity embeddings, indicating successful separation of expression and identity factors. **t-SNE visualizations** revealed distinct clusters for each expression class, and the **confusion matrix** confirmed strong performance across most categories, with minor overlap in similar emotions.

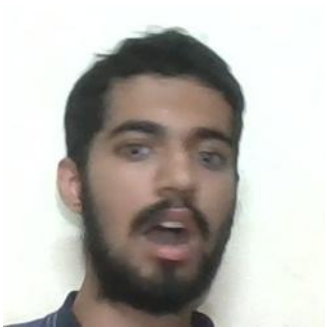
Overall, the results show that DICE effectively reduces identity interference and enhances the robustness of facial expression recognition.



7. Challenges faced

One of the main challenges faced during this project was the creation and manual cleaning of a custom facial expression dataset, as standard FER datasets like CK+ and RAF-DB were inaccessible, and available open-source datasets were too low-resolution (48×48) for our model. We collected around 1800 images, but ensuring their quality required significant manual effort—removing **blurred, out-of-frame, or incorrectly labeled expression images**. Another challenge was forming balanced pairs of images with the same expression for training, which added to the complexity. Additionally, the adversarial training used to minimize mutual information between identity and expression features proved to be sensitive to hyperparameter tuning and occasionally unstable.

Suspect Images



---

8. Conclusion

In this project, we explored and implemented the core ideas of the DICE-FER framework, which aims to decouple identity-specific features from expression representations using mutual information estimation. Due to limitations in accessing standard datasets, we created and manually curated our own facial expression dataset, allowing us to apply the DICE-FER approach in a practical setting. Despite challenges such as data quality, expression pairing, and training instability, the model demonstrated the potential of disentangled representation learning for facial expression recognition. This work not only deepened our understanding of identity-invariant FER but also provided hands-on experience in deep learning, data preprocessing, and adversarial training techniques. The results reinforce the importance of clean data and careful design when applying advanced learning frameworks in real-world scenarios.