YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK – ELEKTRONİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BLM3510 – YAPAY ZEKA

PROJE

Öğrenci No: 21011055, 21011004

Ad-Soyad: Emirhan ÖZSARAY, Yavuz ÇETİN

E-Mail: emirhan.ozsaray@std.yildiz.edu.tr, yavuz.cetin1@std.yildiz.edu.tr

Öğretim Görevlisi: Prof. Dr. Mehmet Fatih AMASYALI

Video Linki: https://youtu.be/yWb-Xv6abRY

Konu Tanıtımı

Projede manga panellerindeki yüz ifadelerinin CNN ve SVM kullanılarak program tarafından tahmin edilmesi amaçlanmıştır. Manga panellerinden alınan yüz ifadeleri 5 ayrı kategoriye göre sınıflandırılmıştır. Bu kategoriler: angry (sinirli), embarrassed (utanmış), happy (mutlu), sad (üzgün), shock (şaşırmış) şeklindedir. Her bir kategoriden 100 tane olmak üzere toplam 500 tane yüz ifadesi toplanmıştır. Bu veri seti test ve train olmak üzere 2 veri setine ayrılmıştır. Test için 100; train için 400 olacak şekilde bu ayrım yapılmıştır. Bu iki veri setindeki yüz ifadeleri yukarıdaki 5 kategoriden eşit miktarda yüz ifadesi içermektedir.

Geliştirme Sürecinde Yaşananlar

Üzgün, utanmış ve sinirli olan yüz ifadeleri birbirleri arasında ayrım yapmanın zor olduğu ifadelerdir. Bu 3 yüz ifadesinin isabet oranı mutlu ve şaşırmış yüz ifadelerine göre daha düşüktür.

Support Vector Machines (SVM)

SVM (Support Vector Machine), görüntü sınıflandırma projelerinde kullanışlı bir makine öğrenimi algoritmasıdır. SVM, verileri sınıflara ayırmak için optimal bir hiperdüzlem oluşturur. Görüntü sınıflandırmasında, SVM, özellik vektörlerini kullanarak görüntüleri farklı sınıflara ayırmak için bir karar sınırı oluşturur. Bu özellik vektörleri, örneğin renk histogramları, kenar yoğunlukları veya özellikli pikseller gibi görüntülerin çeşitli özniteliklerinden elde edilebilir. SVM'nin avantajlarından biri, yüksek boyutlu özellik uzaylarında iyi performans gösterme yeteneğidir. Ayrıca, SVM'nin kernel yöntemi sayesinde, doğrusal olarak ayrılamayan verileri de sınıflandırabilir, bu da görüntülerin karmaşıklığını ele almak için kullanışlıdır.

SVM modelinden yararlanırken grid search kullanılmıştır. Grid search parametreleri aşağıda verilmiştir.

```
# Defining the parameters grid for GridSearchCV
param_grid = {
    'C': [0.1, 1, 10, 100],
    'gamma': [0.0001, 0.001, 0.1, 1],
    'kernel': ['rbf', 'poly']
}
```

```
Best parameters found:  {'C': 0.1, 'gamma': 0.0001, 'kernel': 'poly'}
```

Convolutional Neural Network (CNN)

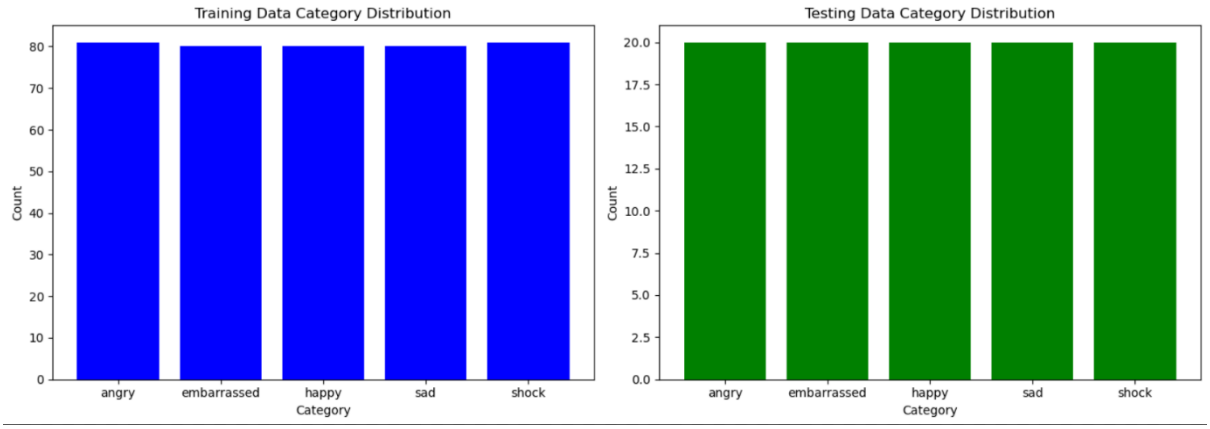
CNN (Convolutional Neural Network), görüntü sınıflandırma projelerinde yaygın olarak kullanılan derin öğrenme modelidir. CNN'ler, özellikle görüntü işleme problemleri için tasarlanmıştır ve insan görsel sisteminden ilham alır. Görüntülerdeki özellikleri tespit etmek için evrişim ve havuzlama katmanlarından oluşan bir yapı kullanır. Bu katmanlar, giriş görüntüsünden farklı ölçeklerde ve özelliklerde öznitelikler çıkararak hiyerarşik bir temsili oluşturur. Ardından, tam bağlantılı katmanlar ve bir sınıflandırıcı, bu öznitelikleri kullanarak görüntüleri belirli sınıflara atar. CNN'lerin avantajlarından biri, özelliklerin elle tanımlanması veya özellik mühendisliği gerektirmemesidir; aksine, öznitelikler doğrudan veriden öğrenilir. Bu, veri setinin karmaşıklığını ele almak ve genelleme yeteneğini artırmak için son derece faydalıdır.

Modelin yapısı aşağıdaki gibidir. 150 epoch boyunca çalıştırılmıştır ve en iyi modelle tahminler yapılmıştır.

```
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(w, h, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(5, activation='softmax')  # 5 output classes
])
```

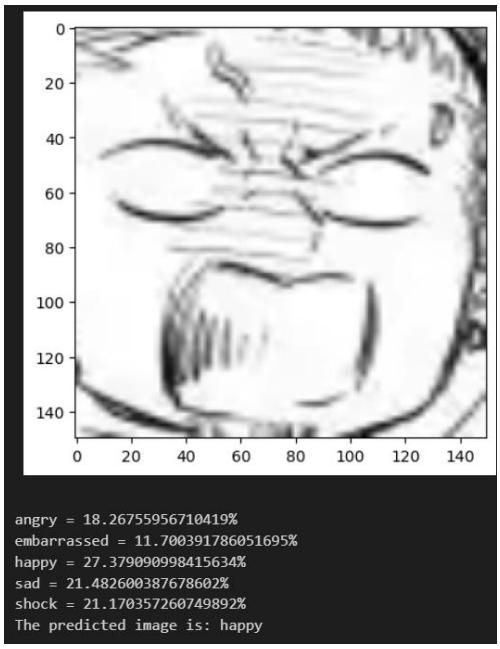
```
history = model.fit(x_train, y_train, epochs=150, validation_data=(x_test, y_test), callbacks=[checkpoint])
```

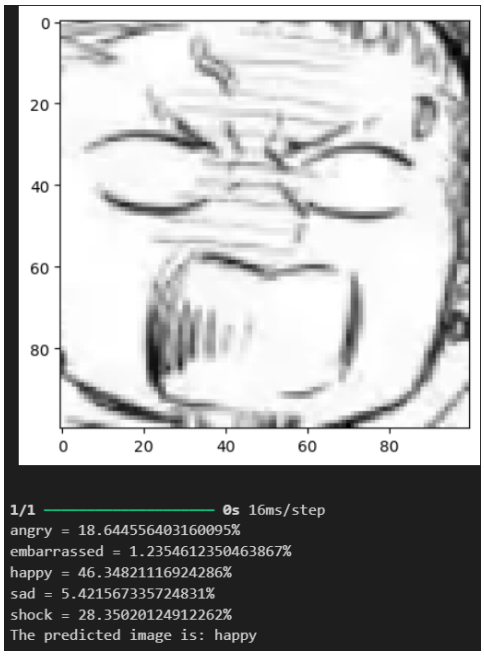
Örnekler



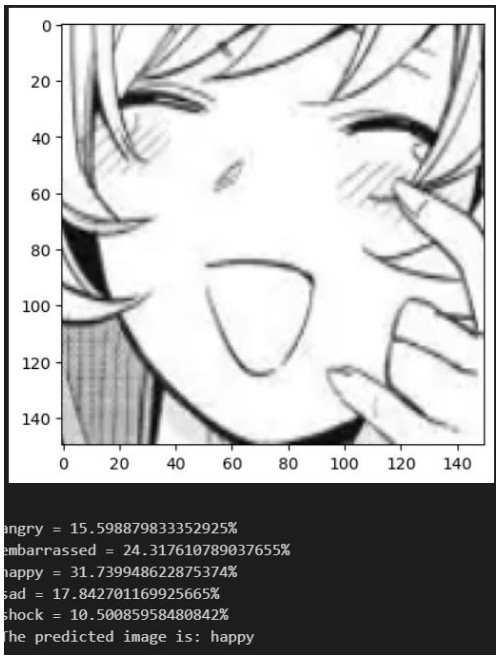
Yukarıdaki grafikte 2'ye ayrılmış veri setlerindeki elemanların kategorilerin sayılarının eşit olduğu görülebilir

Aşağıda gösterilecek olan örneklerde sol örnek SVM, sağ örnek CNN ile tahmin edilmiştir.

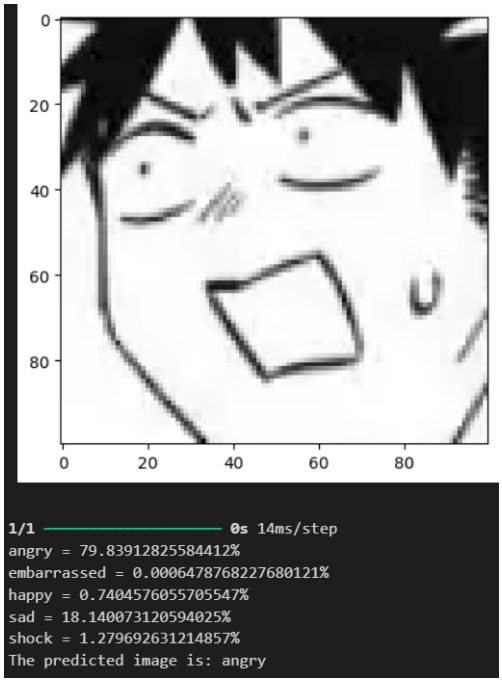




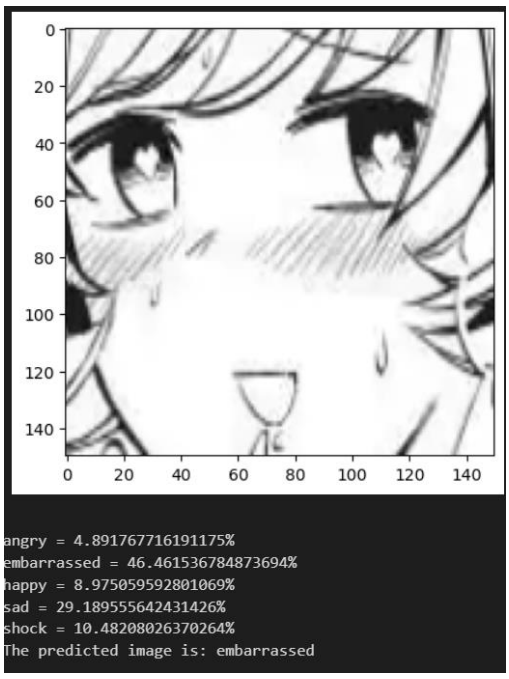
İki modelin de verilen yüz ifadesi örneğini yanlış tahmin ettiği görülmektedir. Bu resimdeki yüz ifadesi açıkça şaşırmış bir yüz ifadesidir.

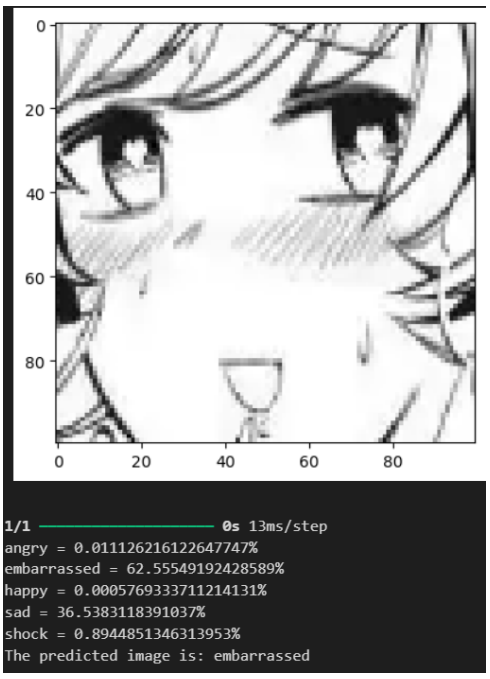


Bu örnekte SVM modeli yüz ifadesini doğru tahmin etmiştir.

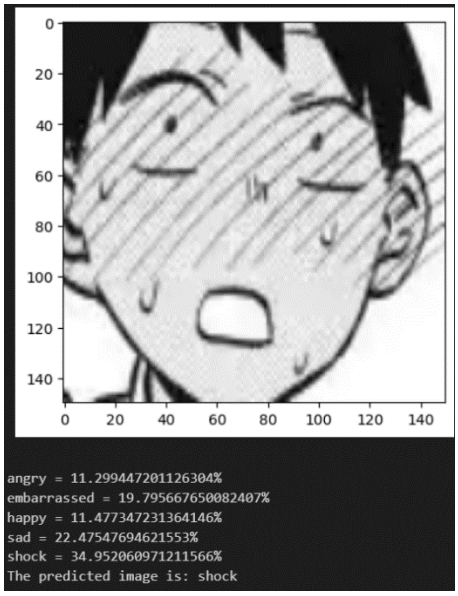


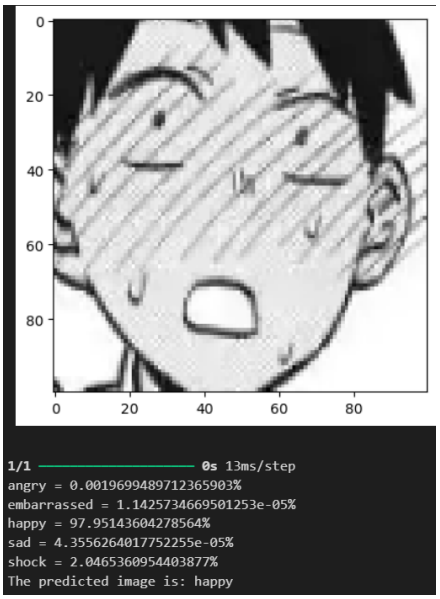
Bu örnekte SVM modeli doğru tahmin etmiştir ama yüz ifadesindeki kaşların çatık olması CNN modelinin sinirli olduğunu tahmin etmesine yol açmıştır.



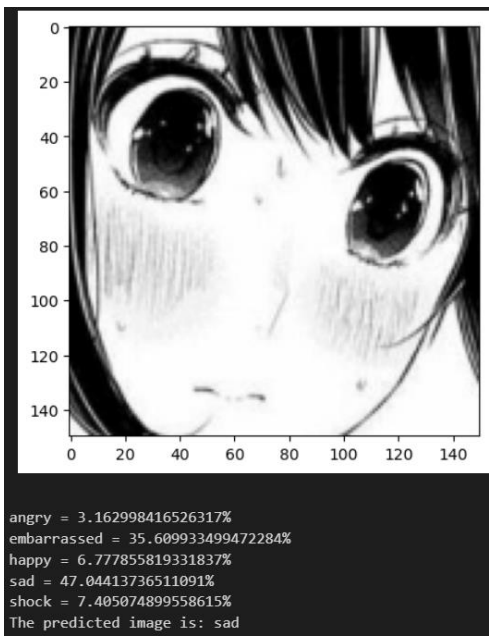


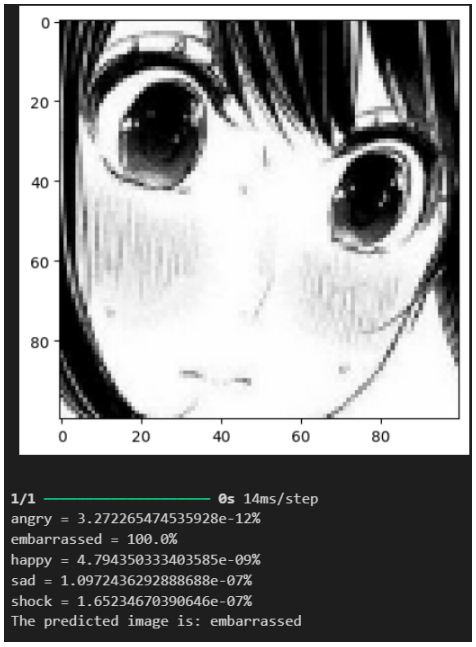
İki model de yüz ifadesini doğru tahmin etmiştir. Verilen puanlara bakılacak olursa utanmış yüz ifadesine verilen yüzdelerin çok fazla olduğu görülecektir. Bu demek oluyor ki iki model de büyük bir farkla tahmin yapmıştır.



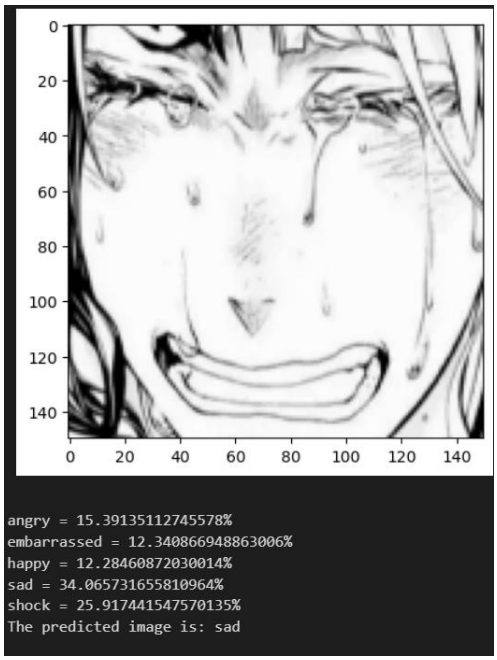


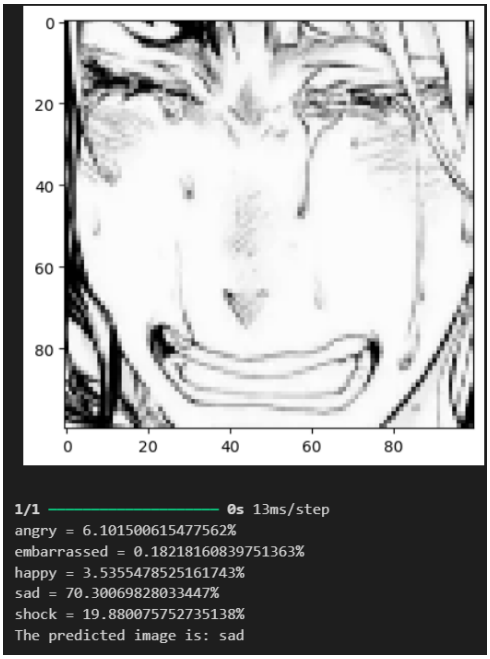
Bu görsel için utanmış ve şaşırmış duyguları tahmin edilmesi beklenmiştir. SVM doğru bir şekilde tahmin yaparken CNN modeli %97'lik bir oranla yanlış ifade olan mutluyu seçmiştir.





Bu yüz ifadesi için hem utanmış hem de üzgün çıkarımları utanmış ağırlıkta olmak üzere yapılabilir. Bu iki model de beklenilen çıktıyı üretmiştir. CNN modeli %100'lük bir oranla utanmış ifadesini bulmuştur. SVM modelinin bulduğu sonuç üzgün ifade olarak bulunmuştur ve utanmış ifade yüzdesi, üzgün ifade yüzdesinin çok yakınındadır. İki model de başarıyla doğru sonuçları bulmuştur.





Bu ifade üzgün bir ifadedir. Her iki model de doğru bulmuştur. Ancak CNN modeli %70'lik oranla üzgün ifadeyi seçerken, SVM modeli yalnızca %34'lük bir oranla üzgün cevabını vermiştir. Bu örnek için CNN modeli daha başarılı olmuştur.

Karşılaştırma

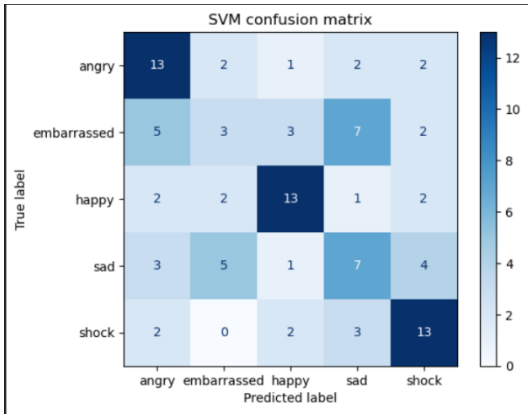
Sistemlerinin başarılı olma derecesi yaptıkların tahminlerin doğruluğuna göre ölçülür.

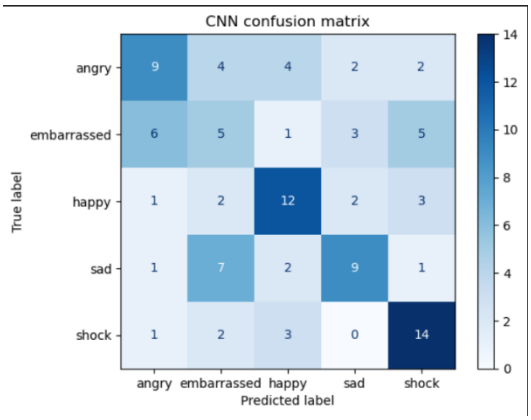
	precision	recall	f1-score	support
angry	0.52	0.65	0.58	20
embarrassed	0.25	0.15	0.19	20
happy	0.65	0.65	0.65	20
sad	0.35	0.35	0.35	20
shock	0.57	0.65	0.60	20
accuracy			0.49	100
macro avg	0.47	0.49	0.47	100
weighted avg	0.47	0.49	0.47	100

4/4 0s 27ms/step
Classification Report:

	precision	recall	f1-score	support
angry	0.50	0.43	0.46	21
embarrassed	0.25	0.25	0.25	20
happy	0.55	0.60	0.57	20
sad	0.56	0.45	0.50	20
shock	0.56	0.70	0.62	20
accuracy			0.49	101
macro avg	0.48	0.49	0.48	101
weighted avg	0.48	0.49	0.48	101

Yukarıda iki modelin de classification report'u verilmiştir. Soldaki SVM, sağdaki CNN'dir. İki modelin embarrassed duygusunu bulurken düşük başarı sergilediği gözlemlenmiştir. Shock ve happy ifadeleri iki modelde de en başarılı olan ifadelerdir. Embarrassed yüz ifadesinde çizilen yüz kızarıklığı, üzgünlükle karıştırılabilir. İki modelin de genel olarak elde ettiği isabet oranı %49'dur.





İki modelin de confusion matrisi verilmiştir. Classification report'ta da görüldüğü gibi happy ve shock duyguları en başarılı olan duygulardır. Bu ikisini angry takip etmekte ve en son en başarısızlar olarak sad ve embarrassed görülmüştür. Confusion matrisindeki bu veriler de üstteki yorumu doğrular niteliktedir. Shock ve happy iki modelde de büyük bir isabet oranı elde etmiştir. Embarrassed duygusunun sad ve angry ile karıştırıldığı bu matriste görülmektedir.

Kaynakça

1. Gotouge Koyoharu, (2016). Kimetsu No Yaiba
2. Tappei Nagatsuki, Shousetsuka ni Narou (2012). Re:Zero Kara Hajimeru Iseaki Seikatsu
3. Yamamoto Souchirou, (2012). Karakai Jouzu no Takagi-san
4. Wakui Ken, (2017). Tokyo Revengers
5. Akasaka Aka, (2015). Kaguya-sama wa Kokurasetai
6. Kotoyama, (2019). Yofukashi no Uta
7. Horikoshi Kouhei, (2014). Boku no Hero Academia
8. Togashi Yoshihiro, (1998). Hunter x Hunter
9. Fukuda Shinichi, (2018). Sono Bisque Doll wa Koi wo Suru
10. Kusano Houki, Kamoshida Hajime (2011). Sakurasou no Pet na Kanojo
11. Watari Masahito, Akatsuki Natsume, (2014). Kono Subarashii Sekai ni Shukufuku wo!
12. Fujimoto Tatsuki, (2018). Chainsaw Man
13. Hagiwara Daisuke, Hero (2011). Horimiya
14. Isayama Hajime, (2009). Shingeki no Kyojin
15. Endou Tatsuya, (2019). Spy x Family
16. https://hako.github.io/dissertation/
17. https://www.researchgate.net/publication/338651794_A_CNN-Based_Tool_to_Index_Emotion_on_Anime_Character_Stickers
18. https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/