

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Video2Manga: Videodan Yarı-Otomatik Manga Üretimi

**PROJE YAZARI**

Ezginur UÇAN

170420037

**DANIŞMAN**

Dr. Öğr. Üyesi Ali SARIKAŞ

**İL, TEZ YILI**

İSTANBUL, 2024



**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Video2Manga: Videodan Yarı-Otomatik Manga Üretimi

**PROJE YAZARI**

Filiz YALÇİN

170419016

**DANIŞMAN**

Dr. Öğr. Üyesi Ali SARIKAŞ

**İL, TEZ YILI**

İSTANBUL, 2024

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencisi …Ezginur Uçan ve Filiz Yalçin… nın “Video2Manga: Videodan Yarı-Otomatik Manga Üretimi” başlıklı bitirme projesi çalışması, 05/07/2024 tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Dr. Öğr. Üyesi ALİ SARIKAŞ (Danışman)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Doç. Dr. Adı SOYADI (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Dr. Öğr. Üyesi Adı SOYADI (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

**İÇİNDEKİLER**

Sayfa

SEMBOLLER LİSTESİ……………………………………………………………………..i

KISALTMALAR LİSTESİ…………………………………………………………………ii

ŞEKİL LİSTESİ……………………………………………………………………………iii

ÖZET……………………………………………………………………………………….iv

BÖLÜM 1. GİRİŞ……………………… ………………………………………………….1

1.1. Bitirme Projesinin Amacı………………………………………………………1

1.2. Literatür Özeti………………………………...………………………………...1

1.2.1. GAN - Generative Adversarial Networks (Çekişmeli Üretici Ağ)......1

1.2.2. Videodan çizgi roman üretimi………………………………………..2

1.2.3.Fotoğraftan manga üretimi……………………………………………3

BÖLÜM 2. MATERYAL VE YÖNTEM …………………………………………………..4

2.1. Materyal………………………………………………………………………..4

2.1.1. Altyazı dosyası…...……..…………...………………………….........5

2.1.2. Karakter tanıma modeli için veri seti.………………………………..5

2.1.3.Manga tarzı çizim için veri setleri ……………………………………7

2.1.3.1.ShapeGAN………………………………………………….7

2.1.3.2.LocationGAN………………………………………………8

2.1.3.3. SizeGAN………………………………………………….10

2.2. Yöntem………………………………………………………………………..11

2.2.1. Key frame seçimi………………..…………………………………..11

2.2.1.1 Konuşma içeren frame’lerin seçimi..…………...………………….11

2.2.1.2 Sahne değişimi içeren frame’lerin seçimi……………….………...12

2.2.2. Layout seçimi……………………………………………………….12

2.2.3. Karakter tanıma modeli…………………………………………….12

2.2.4. Manga tarzında çizim ………………………………………………14

2.2.4.1. GTN (Geometric Transformation Network).......................15

2.2.4.1.1.ShapeGAN………………………………………………15

2.2.4.1.2.Location GAN…………………………………………...17

2.2.4.1.3.SizeGAN………………………………………………...18

BÖLÜM 3. BULGULAR VE TARTIŞMA………………………………………………..19

BÖLÜM 4. SONUÇLAR……………………………………………………………….....22

KAYNAKLAR…………………………………………………………………………….23

EKLER…………………………………………………………………………………….25

ÖZGEÇMİŞ

**SEMBOLLER/SYMBOLS**

: gerçek insan yüzlerinin fotoğraflarından oluşan veri seti

: manga yüzlerinin fotoğraflarından oluşan veri seti

: manga çene landmark veri seti

: gerçek insan çene landmark veri seti

: Shape GAN

: Location GAN

: Location GAN için gerçek insan veri seti

: Location GAN için manga veri seti

: Size GAN

: Size GAN için gerçek insan veri seti

: Size GAN için manga veri seti

**KISALTMALAR/ABBREVIATIONS**

**GAN:** Generative Adversarial Networks (Çekişmeli Üretici Ağ)

**CNN**: Convolutional Neural Network (Evrişimli Sinir Ağları)

**GPU**: Graphics Processing Unit (Grafik İşlem Birimi)

**TPU**: Tensor Processing Unit

**ReLU**: Rectified Linear Unit (Doğrultulmuş Lineer Birim)

**GTN** : Geometric Transformation Network

**ATN**: Appearance Transformation Network (Görünüm Dönüşüm Ağı)

**ŞEKİL LİSTESİ**

Sayfa

Şekil 2.1: Video2Manga mimarisi (a) transkript dosyası yardımıyla videodan frame çıkartma (b) frame’in karakter tanıma modelinden geçirilerek kafa bölgesinin tespit edilmesi ve landmark bilgileri ile shape.txt, location.txt ve size.txt dosyalarının üretilmesi (c) girdi verilerinin , ve ağlarından geçirilerek çıktı üretilmesi (d) üretilen çıktılar ile manga tarzı çizdirme……………………………………………………………4

Şekil 2.1.1. Yüzde bulunan landmark indisleri ……………………………………………9

Şekil 2.2.1. Veri seti etiketleme üzerine örnek …………………………………………....13

Şekil 2.2.2. Modelin karakterleri detect etmesi üzerine test çıktısı………………………..13

Şekil 2.2.3. Karakter tanıma modeline ait kayıp ve performans metrik grafikleri...……...14

Şekil 2.2.4. GTN mimarisi ……………………………………………………………….15

Şekil 2.2.5. Girdi veri setinden alınan verilerin test edilmesi…….……………………....16

Şekil 2.2.6. shapeGAN’ın sahneden alınan bir kesit ile test edilmesi (a) Gerçek imaj (b) Gerçek imajdan elde edilen landmark koordinatlarından Matplotlib ve PCHIP yöntemi ile çenenin çizdirilmesi (c) shapeGAN’ın G\_H2M modelinin ürettiği manga landmark koordinatlarından Matplotlib ve PCHIP yöntemi ile çenenin çizdirilmesi …………..…..17

Şekil 2.2.7. shapeGAN ve locationGAN çıktılarının grafikte birleştirilmesi…..………...18

Şekil 3.1. İlk geliştirilen karakter tanıma modeline ait çıktılar..………………………….19

Şekil 3.2. Gerçek piksel değerleri ve keras ile yapılan eğitim sonucu üretilen landmark’lar……………………………………………………………………………....20

Şekil 3.3: Belirli hareket özelliklerine ait başlangıç bitiş frameleri………………………21

Şekil 3.4: Üretilen shape\_result.txt, location\_result.txt ve size\_result.txt dosyaları kullanılarak çizdirme işlemi sonucu……………………………………………………...21

**ÖZET**

Manga, Japonya kökenli çizgi roman türüne verilen addır. Genellikle siyah-beyaz olarak yayınlanan mangalar, Japon kültüründe önemli bir yer tutar ve dünya çapında geniş bir okuyucu kitlesi tarafından takip edilir. Mangalar, farklı yaş grupları ve ilgi alanlarına yönelik olarak çeşitli türlerde yayımlanır.Video2Manga, gerçek video içeriklerini manga tarzına dönüştürerek sanatsal bir adaptasyon süreci yaratmayı amaçlamaktadır.

Bu çalışma, gerçek video içeriklerinden yarı otomatik olarak manga üretmeyi amaçlayan Video2Manga sistemini tanıtmaktadır. Proje, 2004 yapımı "Harry Potter ve Azkaban Tutsağı" filmindeki "Meet Buckbeak" sahnesini, "Jujutsu Kaisen" tarzında bir mangaya dönüştürmeye odaklanmaktadır. Süreçte, videodaki önemli karelerin seçilmesi ve manga tarzında çizimlere dönüştürülmesi için Generative Adversarial Networks (GAN) kullanılmaktadır.

GAN'lar, birbiriyle rekabet eden iki sinir ağından, bir jeneratör ve bir ayrımcıdan oluşur ve gerçekçi görüntüler üretmek için birlikte eğitilir. Metodoloji, altyazı dosyalarının işlenmesi ile diyalogların tespit edilmesi, karakter tanıma modelleri ile karakterlerin algılanması ve izlenmesi, ve çeşitli GAN mimarileri ile stil transferi süreçlerini içermektedir.

**ABSTRACT**

Manga is the term given to the Japanese-origin comic book genre. Usually published in black and white, mangas hold significant cultural importance in Japan and are followed by a wide global audience. Mangas are published in various genres targeting different age groups and interests.Video2Manga aims to create an artistic adaptation process by transforming real video content into manga style. This project introduces the Video2Manga system, which semi-automatically produces manga from real video content.

The project focuses on transforming a scene from the 2004 film 'Harry Potter and the Prisoner of Azkaban,' specifically the 'Meet Buckbeak' scene, into a manga style resembling 'Jujutsu Kaisen.' The process involves selecting key frames from the video and converting them into manga-style drawings using Generative Adversarial Networks (GANs).

GANs consist of two competing neural networks—a generator and a discriminator—that are trained together to produce realistic images. The methodology includes processing subtitle files to detect dialogues, using character recognition models for character perception and tracking, and employing various GAN architectures for style transfer processes.

# GİRİŞ

Manga, Japon çizgi romanlarına verilen addır. Geçmişten günümüze çokça üretilmiş ve günümüzde dünya çapında sevilen bir kavram hâline gelmiştir. Bu popülerleşme ile mangalar ile ilgili birçok akademik çalışma yapılmıştır.

* 1. **Amaç**

Bu projenin amacı, 2004 yapımı Harry Potter ve Azkaban Tutsağı filminde bulunan “Meet Buckbeak” sahnesindeki karakterlerin Jujutsu Kaisen adlı manga tarzı ile çizdirilmesi ve sonuçların “mangalaştırılmasıdır.” Bu sayede gerçek insanların bulunduğu bir videodan otomatik olarak manga üretimi konusunda bir adım atılması amaçlanmıştır.

* 1. **Literatür Özeti**

Videodan mangaya direkt bir çeviri yapılan çalışmaya rastlanmamıştır, ancak buna benzer çalışmalar taranıp birkaç başlık altında toplanmıştır.

* + 1. **GAN - Generative Adversarial Networks (Çekişmeli Üretici Ağ)**

Çekişmeli Üretici Ağlar (Generative Adversarial Networks - GAN), yapay zekâ ve makine öğrenmesi alanında son yıllarda büyük ilgi gören ve etkili sonuçlar üreten bir derin öğrenme mimarisidir. GAN, Ian Goodfellow ve meslektaşları tarafından 2014 yılında tanıtılmıştır [1] ve iki ana bileşenden oluşur: *üretici (generator)* ve *ayrımcı (discriminator).* Bu iki bileşen birbirlerine karşıt (adversarial) bir şekilde eğitilir.

Üretici, rastgele bir girdi vektöründen başlayarak gerçekçi veriler oluşturmayı amaçlar. Ayrımcı ise, gerçek veriler ile üreteç tarafından oluşturulan sahte verileri ayırt etmeye çalışır. Bu iki model, oyun teorisi çerçevesinde bir minimaks oyununda olduğu gibi eğitilir. Üretici, ayrımcının sahte verileri doğru bir şekilde tespit edemeyeceği kadar gerçekçi veriler üretmeye çalışırken, ayrımcı da sahte ve gerçek verileri doğru bir şekilde ayırt edebilmek için kendini geliştirir. Bu çekişmeli süreç, her iki bileşenin de performansını artırır ve sonuçta üretici, oldukça gerçekçi veriler üretebilir hale gelir.

GAN'in başlıca uygulama alanları arasında görüntü oluşturma ve işleme, video üretimi, veri artırma, ses sentezi ve doğal dil işleme yer almaktadır. Özellikle yüksek çözünürlüklü görüntülerin üretilmesi, stil transferi, süper çözünürlük gibi görsel görevlerde olağanüstü sonuçlar elde edilmiştir.

* + 1. **Videodan çizgi roman üretimi**

Won-Il Hwang ve arkadaşları [2], video akışlarından yarı otomatik bir şekilde çizgi roman akışlarının oluşturulmasını kolaylaştırmak için tasarlanmış bir sistem olan CORVIS'i (COmics Rendering system on VIdeo Stream) geliştirmişlerdir. CORVIS’i oluşturan adımlar; videodaki önemli sahnelerin manuel seçimi, sahneleri basitleştirilmiş çizimlere dönüştürmek için Mean-Shift segmentasyonu, hız çizgileri ve dönme yörüngeleri gibi stilize çizgi roman efektlerinin eklenmesi şeklindedir. Ek olarak sistem, çizgi roman kesimlerinde ses efektleri için diyalogları ve yankılı kelimeleri temsil etmek için otomatik olarak kelime balonları eklemiştir.

Dong-Sung Ryu ve arkadaşları [3], video akışlarını siyah/beyaz çizgi romanlara dönüştürmek için tasarlanmış yarı otomatik bir sistem olan CINETOON'u geliştirmişlerdir. Sistem dört ana bileşenden oluşur: Key Frame Çıkartıcı, Görüntü Tabanlı Çizgi Roman Kesitine Dönüştürücü, Arka Plan Efekti Stilizatörü ve Çizgi Roman Elementleri Oluşturucu. Frame görüntüleri için yeni bir siyah/beyaz ikilileştirme algoritması, ön plan nesne segmentasyonu için basit bir algoritma ve bursting, speed lining ve trembling gibi stilistik efektler için parametreli bir font sistemi gibi yenilikçi teknikler kullanmışlardır.

Meng Wang ve arkadaşları [4], iki ana ilkeyi izleyerek film kliplerini otomatik olarak çizgi romanlara dönüştüren Movie2Comics adlı bir şema önermektedir: bilgi korumasını optimize etmek ve çizgi roman stillerinde çıktılar üretmek. Şema üç bileşenden oluşur: script (senaryo) - yüz eşleme, açıklayıcı frame çıkarma ve karikatürleştirme. Script - yüz eşleme, karakterlerin yüzlerini script dosyalarına bağlamak için yüz izlemeyi (face tracking) kullanırken, açıklayıcı frame çıkarma sonuç görseli için frame’ler oluşturur. Karikatürleştirme süreci, panel ölçeklendirme, stilizasyon ve layout tasarımını içerir.

Guangmei Jing ve arkadaşları [5], konuşma içeren videoları manga tarzı bir düzen ile çizgi romanlara dönüştürmek için yeni bir yaklaşım sunmuştur. Yöntem, bir çizgi roman sayfasındaki paneller ve kelime balonları gibi temel bileşenleri akıllıca düzenleyerek içerik odaklı manga tarzı bir düzen elde etmeye odaklanır. Kelime balonlarını ilgili konuşmacıların yanına yerleştirmek için konuşmacı algılama tekniklerini kullanır ve ortaya çıkan çizgi roman sayfasının görsel çekiciliğini ve bilgi içeriğini geliştirir. Yaklaşım, nihai düzeni verimli bir şekilde belirlemek için önceden tanımlanmış bir ölçüyü en üst düzeye çıkaran bir optimizasyon sürecini içerir. Ek olarak, bu optimizasyon görevi için uyarlanmış verimli bir Markov zinciri Monte Carlo (MCMC) örnekleme algoritması sunmaktadır.

Maciej Pęśko ve arkadaşları [6], Çekişmeli Üretici Ağlara (GAN) dayalı bir nöral stil algoritması kullanarak videoları çizgi romanlara dönüştürmek için Comixify adlı bir yöntem önermiştir. İşlem iki aşamadan oluşur: key frame çıkarma ve stil aktarımı. Key frame’ler, videoyu etkili bir şekilde özetlemek için vurgu skorlarını değerlendiren bir algoritma kullanılarak seçilir. Stil aktarım bölümü, seçilen key frame’leri çizgi roman formatına dönüştürmek için son teknoloji bir stil aktarım çözümü kullanır. Ham videoları girdi olarak alır ve web tabanlı bir uygulama ile tamamen biçimlendirilmiş çizgi roman çıktıları alır.

* + 1. **Fotoğraftan manga üretimi**

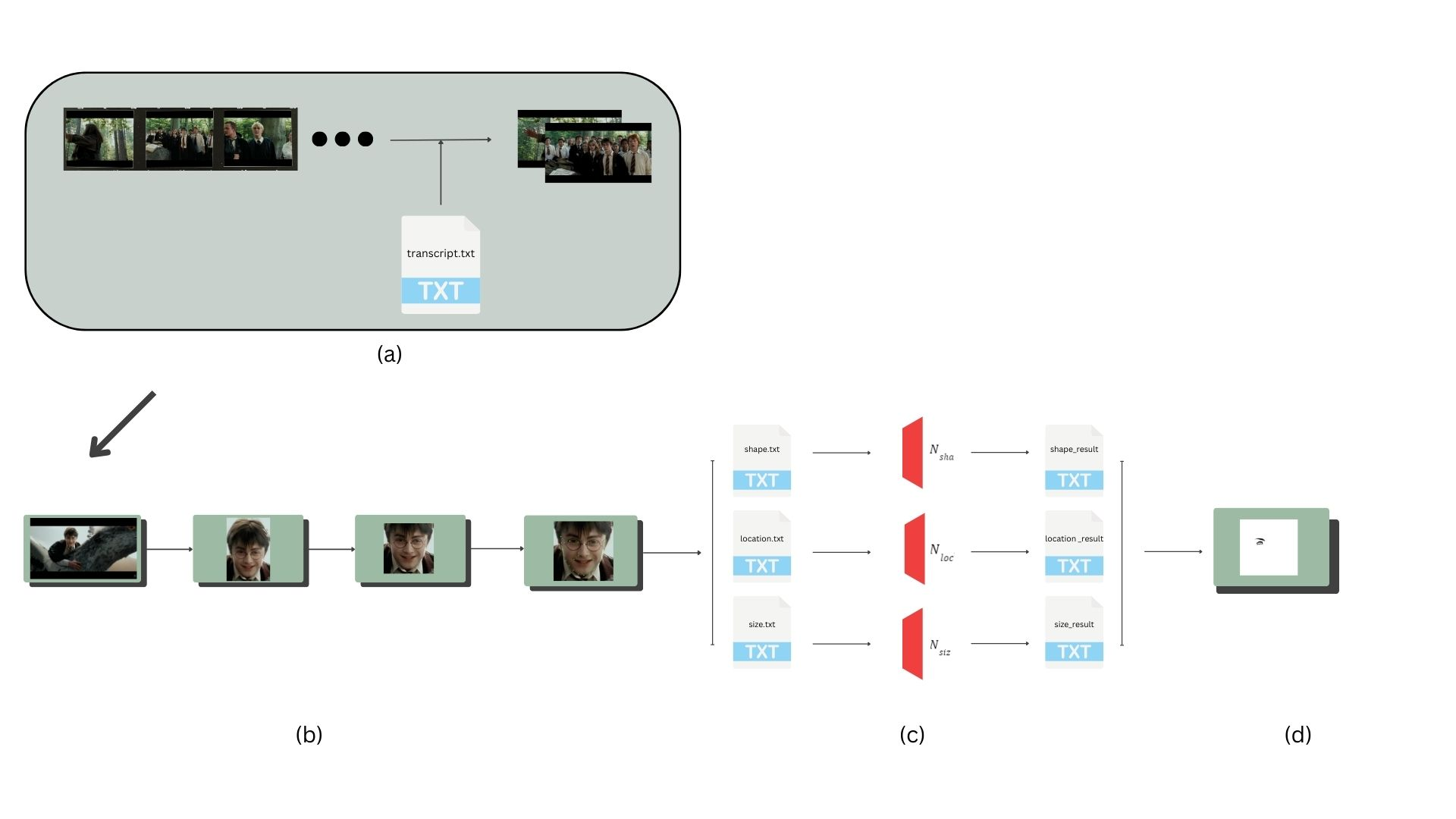
Yang Chen ve arkadaşları [7], gerçek dünya fotoğraflarını yüksek kaliteli çizgi film tarzı görüntülere dönüştürmek için bir Çekişmeli Üretici Ağ (GAN) framework’ü olan CartoonGAN'ı önermiştir. Yöntem, eğitim için eşleştirilmemiş fotoğrafları ve çizgi film görüntülerini kullanır ve iki yeni kayıp sunar: stil varyasyonlarını işlemek için anlamsal bir içerik kaybı ve net kenarlar için kenarı teşvik eden bir çekişmeli kayıp. Ek olarak, ağ yakınsamasını geliştirmek için bir başlatma aşaması uygulanır.

Hao Su ve arkadaşları [8], yüz fotoğraflarını manga stiline dönüştürmeyi amaçlayan, eşleştirilmemiş fotoğraftan mangaya çeviri için yeni bir Çekişmeli Üretici Ağ (GAN) tabanlı yöntem olan MangaGAN'ı tanıtmıştır. MangaGAN, geometrik özellikler oluşturmak ve yüz bölgelerini manga alanına dönüştürmek için deneyimli manga sanatçılarının çizim sürecinden ilham alan özel bir çoklu GAN mimarisinden yararlanır. Yüksek kaliteli manga yüz üretimi sağlamak için yöntem, kontur çizgilerini iyileştirmek için yapısal bir yumuşatma kaybı ve etki alanı benzerliğini artırmak için benzerlik koruma modülü içerir. MangaGAN, hem yüz benzerliğini hem de manga stilini korumayı amaçlıyor.

# MATERYAL VE YÖNTEM

Videodan mangaya çevrim için videonun birçok aşamadan geçmesi gereklidir. Bu çalışmada uygulanan adımlar; key frame seçimi, layout seçimi ve frame’lerin manga panellerine yerleştirilmesi, karakter tanıma modeli, manga tarzında çizim ve baloncuk yerleştirme olarak belirlenmiştir.

Videodan manga üretiminde ilk olarak yapılması gereken adım, manga sayfalarına eklenecek sahnelerin belirlenmesidir. Bu sahneler belirlenirken manganın videoyu iyi bir şekilde temsil edebilmesi için farklı çalışmalarda farklı kriterler uygulanmıştır. Konuşma içeren sahneler videodaki akışın anlaşılması için kritik bir önem taşıyacağından karakterlerin konuştuğu sahnelerin seçilmesi gerektiği sonucuna varılmıştır. Mangalarda konuşma içermese de sahne değişimi bulunan panellerin bulunması sebebiyle videoda da benzer şekilde sahne değişimi bulunan kısımların oluşturulacak mangaya eklenmesi gerektiği düşünülmüştür (bkz. EK 1 ve EK 2).



Şekil 2.1: Video2Manga mimarisi (a) transkript dosyası yardımıyla videodan frame çıkartma (b) frame’in karakter tanıma modelinden geçirilerek kafa bölgesinin tespit edilmesi ve landmark bilgileri ile shape.txt, location.txt ve size.txt dosyalarının üretilmesi (c) girdi verilerinin , ve ağlarından geçirilerek çıktı üretilmesi (d) üretilen çıktılar ile manga tarzı çizdirme

* 1. **Materyal**
     1. **Altyazı dosyası**

Mangaya çevrilmesi planlanan video[[1]](#footnote-0) YouTube üzerinden elde edilmiştir. Bu video altyazıya sahip olduğundan altyazı dosyası YouTube’un “Transkripti Göster” özelliği kullanılarak elde edilebilmiştir. Transkript kopyalandıktan sonra txt formatında bir dosyaya yapıştırılmıştır. Dosyada her bir konuşma içeren metin, videoda gerçekleşme süresine göre etiketli şekilde elde edilmiştir. Bu etiketlerin başına manuel olarak konuşmacı karakterin ismi de eklenmiştir. Transkriptteki format aşağıdaki şekildedir:

| 0:01  [hagrid]TAD-DA-DA-DAAA!  0:08  [hagrid]Isn't he beautiful?  0:12  [hagrid]Say hello to Buckbeak.  0:15  [ron]Hagrid.  0:15  [ron]Exactly what is that?  … |
| --- |

* + 1. **Karakter tanıma modeli için veri seti**

OpenCV kütüphanesi yardımıyla ilgili videoda 30 fps ile örnekleme yapılıp toplam 7838 adet frame elde edilmiştir. Bu frame’ler görüntü netliği daha yüksek olanlar tercih edilerek 2670 tanesi karakter-script eşleştirmesi yapmak üzere eğitilecek model için veri setine eklenmiştir.

Elde edilen veri setini etiketleme için Roboflow [9] aracı kullanılmıştır. Roboflow, nesne tespiti, sınıflandırma ve segmentasyon gibi bilgisayarlı görü görevlerinde kullanılmak üzere tasarlanmış, geniş kapsamlı ve kullanıcı dostu bir platformdur. Bu araç, araştırmacıların ve geliştiricilerin veri setlerini hızlı ve verimli bir şekilde etiketlemelerine olanak tanır. Roboflow'un sunduğu özellikler arasında otomatik etiketleme, veri artırma (augmentation) teknikleri, ve veri setlerini farklı formatlarda dışa aktarma gibi işlevler bulunmaktadır. Ayrıca, platformun bulut tabanlı yapısı sayesinde kullanıcılar verilerini herhangi bir cihazdan erişebilir ve işleyebilirler. Etiketleme sürecinin hızlandırılması ve doğruluğun artırılması amacıyla Roboflow, yapay zeka destekli araçlar sunar. Bu sayede, büyük ve karmaşık veri setleri bile kısa sürede ve yüksek doğrulukla etiketlenebilir. Roboflow'un kullanıcı dostu arayüzü ve kapsamlı dokümantasyonu, kullanıcıların hızlı bir şekilde adapte olmalarını ve platformun tüm özelliklerinden faydalanmalarını sağlar. Bu nedenle, Roboflow, veri etiketleme süreçlerinde etkinliği ve verimliliği artırmak için bu çalışmada tercih edilmiştir.

Karakter tanımasını yapacak modelin eğitimi için Google Colab [10] ortamı kullanılmıştır. Google Colab, herhangi bir kurulum gerektirmeyen ve GPU ile TPU gibi güçlü bilgi işlem kaynaklarına ücretsiz erişim sağlayan bulut tabanlı bir Jupyter not defteri hizmetidir. Bu platform, özellikle derin öğrenme ve makine öğrenmesi projeleri için ideal bir ortam sunar. Kullanıcılar, Python kodlarını doğrudan tarayıcı üzerinden çalıştırabilir ve güçlü hesaplama kaynaklarına erişebilirler. Google Colab'in sunduğu en önemli avantajlardan biri, GPU ve TPU desteği sayesinde model eğitimi sürecinin önemli ölçüde hızlandırılmasıdır. Ayrıca, belirli bir süreye kadar ücretsiz GPU desteği sunması, araştırmacıların ve geliştiricilerin maliyet etkin bir şekilde projelerini yürütmelerine olanak tanır. Google Colab'in kullanım kolaylığı, kullanıcı dostu arayüzü ve kapsamlı dokümantasyonu, platformun hızlı bir şekilde öğrenilmesini ve kullanılmasını sağlar. Ayrıca, Google Colab'in entegre olduğu Google Drive ile veri saklama ve paylaşma işlemleri de oldukça kolaydır. Bu özellikler, Google Colab'in karakter tanıma modeli eğitimi gibi hesaplama yoğun işlemler için tercih edilmesinde önemli bir rol oynamıştır.

Eğitim için YOLO modelinin 8.0.20 [11] versiyonu tercih edilmiştir. YOLO (You Only Look Once), konvolüsyonel sinir ağları (CNN) kullanarak nesne tespiti yapan bir derin öğrenme algoritmasıdır. YOLO algoritması, nesne tespitini tek bir ileri geçiş (forward pass) ile gerçekleştirir, bu da onu hızlı ve verimli bir seçenek haline getirir. Bu model, bir görüntüyü grid yapısına böler ve her bir grid hücresi içinde belirli sayıda nesne için sınırlayıcı kutular (bounding boxes) ve bunların sınıf olasılıklarını tahmin eder. YOLO'nun temel avantajlarından biri, gerçek zamanlı nesne tespiti yapabilme yeteneğidir; bu nedenle, özellikle düşük gecikme süresi gerektiren uygulamalarda tercih edilir. YOLO 8.0.20 versiyonu, daha önceki sürümlere kıyasla geliştirilmiş doğruluk ve hız sunan güncellenmiş mimariler ve eğitim teknikleri içerir. Bu sürüm ayrıca, daha iyi genel performans ve esneklik sağlayan çeşitli veri artırma (augmentation) stratejilerini ve hiperparametre ayarlarını destekler. YOLO'nun bu son sürümünün seçilmesinin bir diğer nedeni, geniş kullanıcı topluluğu ve kapsamlı dokümantasyonu sayesinde geliştiricilere ve araştırmacılara modelin entegrasyonu ve uyarlanması konusunda geniş destek sağlamasıdır. YOLO'nun bu özellikleri, onu çeşitli bilgisayarlı görü görevlerinde, özellikle de nesne tespiti uygulamalarında vazgeçilmez bir araç haline getirmektedir.

* + 1. **Manga tarzı çizim için veri setleri**

Manga tarzı çizim aşamasında kullanılmak üzere gerçek insan yüzlerinin fotoğraflarından oluşan bir veri seti () ve manga yüzlerinin görsellerinden oluşan bir veri seti () elde edilmiştir.

veri setini elde etmek için HuggingFace platformunda bulunan Humanface\_of\_various\_age\_groups veri seti [13] kullanılmıştır. Bu çalışmada yalnızca ön yüz görsellerinin manga tarzında çizdirilmesi amaçlandığı için bu veri setinden alınan görsellerde Nafaf Alageel’in Face Pose Detection [14] adlı çalışmasından yardım alınarak yüz pozisyonu tespiti yapılıp yalnızca ön yüz görselleri elde edilmiştir. Yan profil görselleri bu çalışmanın konusu dışında bulunduğundan hariç bırakılmıştır. Toplam 726 adet görsel elde edilmiştir.

veri setini elde etmek için son zamanlarda popülaritesi ve hayran kitlesinin hızla artması sebebiyle Crunchyroll Anime Ödülleri’nde 2024 yılının en iyi animesi ödülünü alan Jujutsu Kaisen adlı animenin aynı isimdeki mangası seçilmiştir [15]. İlgili mangadan yüz görselleri toplanmıştır. Manuel olarak kırpma, rotasyon işlemlerine tabi tutulmuştur. Roboflow’dan dışa aktarılan tüm veri setlerinde 256x256

* + - 1. **ShapeGAN**

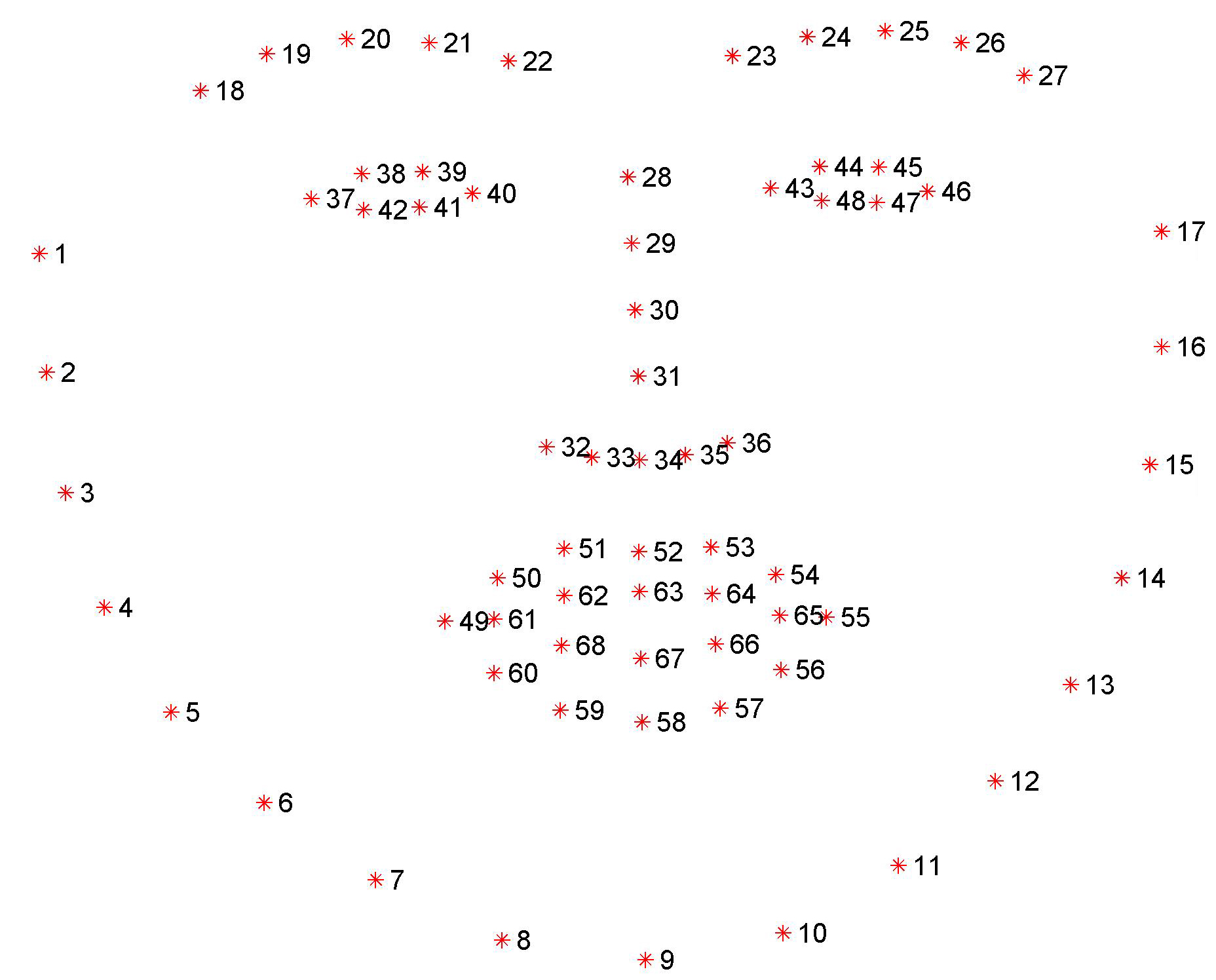
Çene bölgesini çevirmek için bir GAN modeli () yazılmıştır. Bu modelin eğitiminde insan çenesine ait landmark’ların bulunduğu bir insan veri seti () ve manga çenesine ait landmark’ların bulunduğu bir manga veri seti () kullanılmıştır.

için, HuggingFace’ten elde edilen veri setindeki görseller üzerinde dlib kütüphanesi [16] yardımıyla yüzdeki çene landmark’larının tespiti yapılmıştır. Dlib kütüphanesinde çene bölgesine denk gelen noktalar, 0 - 16 indislerine sahip 17 noktaya karşılık gelmektedir. Elde edilen 17 adet çene landmark’ının piksel değerleri normalize edilerek elde edilen anotasyonlar 34 satırda, her bir satırda bir değer olacak şekilde (, , , , …, , ) txt formatında kaydedilmiştir. Elde edilen 726 anotasyon dosyası için girdi veri seti olarak belirlenmiştir.

için öncelikle Roboflow ortamında bir keypoint detection projesi oluşturulmuştur. Bu projeye ön yüz bulunan manga görselleri aktarılarak 0-16 indislerine sahip yüz landmark’ları manuel olarak etiketlenmiştir. Toplam 99 adet manga görseli projedeki veri setine eklenmiştir. Veri seti YOLOv8 formatında zip dosyası olarak dışa aktarılmıştır. YOLOv8 formatında her bir satırda; aralarında bir boşluk olacak şekilde sırasıyla: Sınıf numarası, merkezin x koordinatı, merkezin y koordinatı, bounding box genişliği, bounding box yüksekliği, , , , , , , , , , … (i = 0, 1, …, 16) şeklinde gitmektedir. Burada , noktaları x ve y koordinatlarını ifade ederken, ise görünürlük (visibility) bayrağıdır. Bu formattan, veri setiyle aynı formata dönüşüm için 17 çene noktasının x ve y koordinatları alınıp 34 satırdan oluşan anotasyon dosyaları elde edilmiştir. Elde edilen 99 adet anotasyon dosyası için girdi veri seti olarak belirlenmiştir.

* + - 1. **LocationGAN**

Göz, ağız ve burun bileşenlerinin konumlarının çevrilmesi için bir GAN modeli () yazılmıştır. Bu modelin eğitiminde ilgili bileşenlerin çenenin sol uç noktası, sağ uç noktası ve dip noktasına olan uzaklıklarının normalize edilmiş değerlerinden oluşan bir insan veri seti () ve mangada ilgili bileşenlerin çenenin sol uç noktası, sağ uç noktası ve dip noktasına olan uzaklıklarının normalize edilmiş değerlerinden oluşan bir manga veri seti () kullanılmıştır.



Şekil 2.1.1: Yüzde bulunan landmark indisleri [17]

için yüz bileşenlerinin kesin koordinatlarının değil de çenenin sol uç noktası, sağ uç noktası ve dip noktasına olan uzaklıklarının veri seti olarak kullanılması kararlaştırılmıştır. Bunun sebebi, yüz bölgesinin dışına çıkan sonuçlar elde etmemek içindir. HuggingFace’ten elde edilen veri setindeki görseller üzerinde dlib kütüphanesi yardımıyla öncelikle yüzdeki çene bölgesinde sol uç nokta, sağ uç nokta ve dip nokta (çene ucu) koordinatları elde edilmiştir. Tespit edilen 68 landmark noktasından sırasıyla 0, 8 ve 16. indisler bu noktalara karşılık gelmektedir. Daha sonra yüzdeki bileşenlerin orta noktasının bulunabilmesi için sol ve sağ uç noktaları tespit edilmiştir, bu noktalara karşılık gelen indisler; sol göz için 36 ve 39, sağ göz için 42 ve 45, burun için 31 ve 35, ağız için ise 48 ve 54. indislerdir. Görsel 2.1.1’de görülen indisler 1’den başlatılmıştır, dlib kütüphanesinde 0’dan başlamaktadır. Elde edilen landmark bilgilerinden her bir bileşenin orta noktasının koordinatları tespit edilmiştir. Orta noktalar ile sol ve sağ uç noktaya olan uzaklık x ekseni dikkate alınarak hesaplanırken, dip noktaya olan uzaklık y ekseni dikkate alınarak hesaplanmıştır ve yalnızca bu değerler metin dosyasına aktarılmıştır. Toplam 725 adet metin dosyası, için girdi verisi olarak belirlenmiştir. Metin dosyasının formatı aşağıdaki şekildedir:

| el\_cl el\_cr el\_cb  er\_cl er\_cr er\_cb  n\_cl n\_cr n\_cb  m\_cl m\_cr m\_cb |
| --- |

* el\_cl: sol gözün çenenin sol uç noktasına olan uzaklığının normalize edilmiş x değeri
* el\_cr: sol gözün çenenin sağ uç noktasına olan uzaklığının normalize edilmiş x değeri
* el\_cb: sol gözün çenenin dip noktasına olan uzaklığının normalize edilmiş y değeri

er (sağ göz), n (burun) ve m (ağız) için de benzer şekilde tanımlama yapılabilir.

için, Roboflow ortamında oluşturulan keypoint detection projesine 4 adet sınıf daha eklenmiştir: left-eye, right-eye, nose ve mouth. Veri setinde bulunan 99 görselden 85 tanesinde ilgili yüz bileşenlerinin landmark’ları manuel olarak etiketlenmiştir. Veri seti YOLOv8 formatında zip dosyası olarak dışa aktarılmıştır. Dosya içinde her satırda bulunan Sınıf Numarası kontrol edilerek, için yapılan işlem burada da yapılmıştır. Örneğin sol göz için Sınıf Numarası 1 olduğundan 1 ile başlayan satır bulunur, burada Sınıf Numarası hariç 4, 5 ve 13, 14. indisler gözün sol uç noktası ile sağ uç noktasının x, y noktalarını ifade eder. Benzer şekilde çene, sağ göz, burun ve ağız landmark’ları için de değerler bulunabilir ve hesaplama yapılabilir. Değerler normalize şekilde bulunduğu için normalize işlemine tabi tutulmaz. Elde edilen değerler ile aynı formatta metin dosyalarında saklanır.

* + - 1. **SizeGAN**

Göz, ağız ve burun bileşenlerinin boyutlarının çevrilmesi için bir GAN modeli () yazılmıştır. Bu modelin eğitiminde ilgili bileşenlerin genişliklerinin normalize edilmiş değerlerinden oluşan bir insan veri seti () ve mangada ilgili bileşenlerin genişliklerinin normalize edilmiş değerlerinden oluşan bir manga veri seti () kullanılmıştır.

için, daha önce veri setini oluştururken kullanılan yöntemle, bileşenlerin sol ve sağ uç noktalarının x ve y değerleri bulunur. Bu noktaların x ekseninde farkı alındığında normalize edilmiş genişlik bilgisi elde edilir. Elde edilen değerler 4 satırdan oluşacak şekilde metin dosyası olarak kaydedilir. Toplam 725 adet metin dosyası, için girdi verisi olarak belirlenmiştir.

için, daha önce veri setini oluştururken kullanılan yöntemle, bileşenlerin sol ve sağ uç noktalarının x ve y değerleri bulunur. Bu noktaların x ekseninde farkı alındığında normalize edilmiş genişlik bilgisi elde edilir. Elde edilen değerler 4 satırdan oluşacak şekilde metin dosyası olarak kaydedilir. Toplam 85 adet metin dosyası, için girdi verisi olarak belirlenmiştir.

* 1. **Yöntem**
     1. **Key frame seçimi**

Key frame seçilirken iki adet kriter belirlenmiştir: Konuşma içeren frame’lerin seçimi ve sahne değişimi içeren frame’lerin seçimi.

**2.2.1.1. Konuşma içeren frame’lerin seçimi**

Video2Comics [4] çalışmasında olduğu gibi, konuşan-keyframe seçimi yapılabilmesi için, öncelikle karakter tanıma yapacak bir model eğitildi. Konuşan-keyframe tanımı konuşma içeren sahneleri ifade etmektedir.

Videonun transkript dosyasında bulunan zaman damgalarından yararlanılarak konuşma içeren sahnelerin isimlerinin belirlenmesi sağlandı. (1) numaralı eşitlikte belirtildiği gibi saniye cinsinden süre ile örnekleme oranı (fps: frame per second) değerleri çarpılarak frame ismi elde edildi.

time\_seconds = minutes × 60 + seconds

frame\_number = round(time\_seconds × frame\_rate) (1)

Elde edilen frame isimlerine karşılık gelen görüntülere, daha önce videodan 30 fps ile örnekleme yöntemiyle elde edilen veri setinden ulaşıldı.

**2.2.1.2. Sahne değişimi içeren frame’lerin seçimi**

Sahne değişimi tespitinde Movie2Comics [4] çalışmasında izlenen yönteme benzer bir yol izlenmiştir. Videodaki sahneler subshot’lara ayrıldıktan sonra kamera hareketleri, (2) eşitliğinde ifade edildiği gibi birtakım parametreler ile temsil edilip her bir subshot *static, pan (yatay), tilt (dikey), rotation (dönme), zoom (yakınlaştırma/uzaklaştırma),* ve *object motion* kategorilerine ayrılabilir [18]. Burada , , …, hareket (motion) parametrelerini ifade etmektedir. İki frame arasında Lucas-Kanade metodu ile optik akış hesaplaması yapılır ve hareket parametreleri elde edilir. Her bir frame için , , , ve parametreleri hesaplanır ve frame’in kategorisi en yüksek değere sahip parametreye göre belirlenir. Hesaplanan parametrelerin birbirine eşit olması durumu *static* olarak kategorize edilmektedir.

(2)

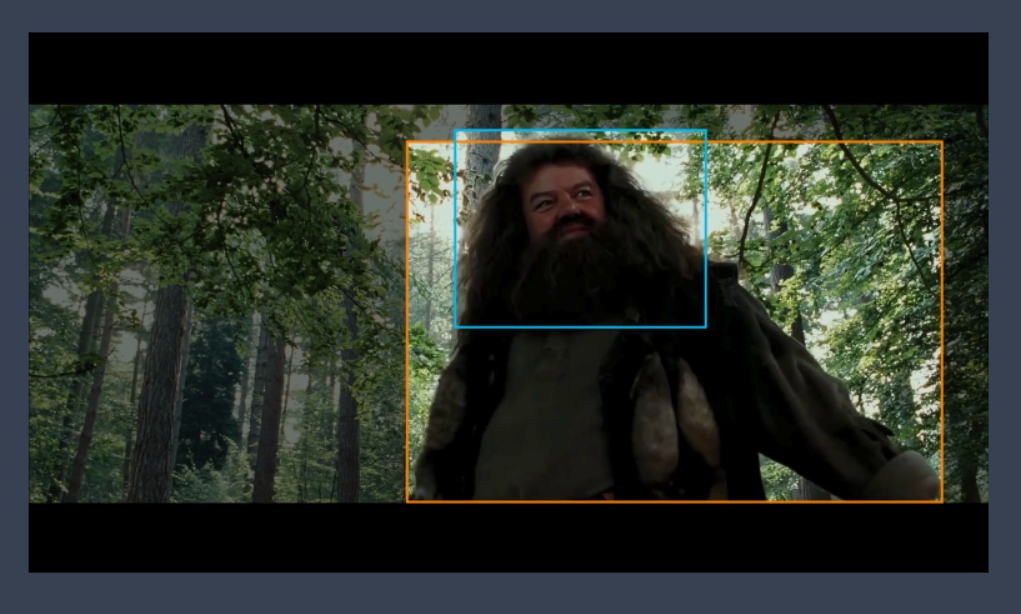
Her bir çekim segmenti, aynı kategorideki diğer segmentlerle birleştirilmek üzere benzer kategorilerde gruplanır. Daha sonra her bir grup için başlangıç ve bitiş frame’lerini belirleyen bir algoritma oluşturulmuştur. Sahne değişimi içeren frame’lerin seçimi, başlangıç ve bitiş frame’leri baz alınarak gerçekleştirilir.

* + 1. **Layout seçimi**

Manga çiziminde kullanılan birçok hazır layout şablonu incelenmiştir. Bu şablonlardan bir tanesi default bir şekilde seçilerek projenin bu aşamasında kullanılmıştır. Yani layout seçimi aşamasında dinamik bir yaklaşımdan ziyade statik bir yaklaşıma gidilmiştir.

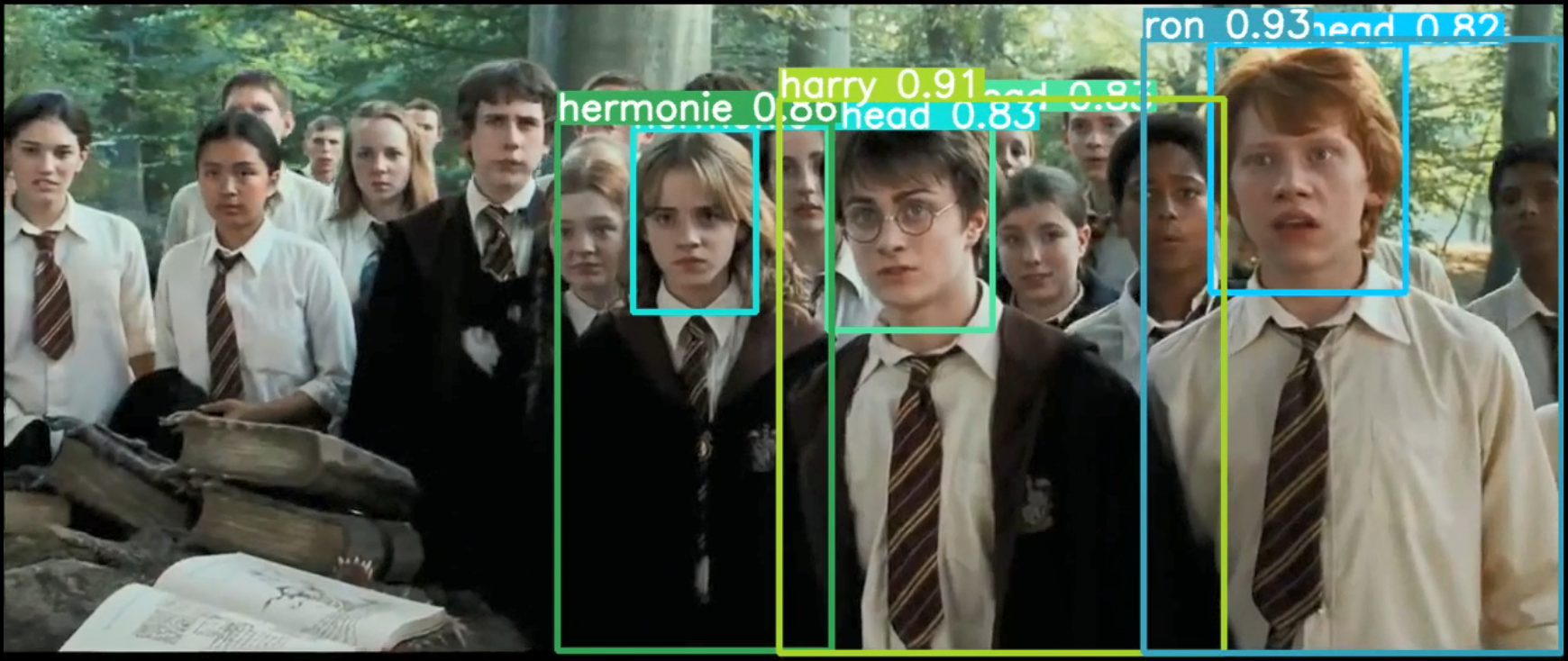
* + 1. **Karakter tanıma modeli**

Karakterlerin yüzlerini manga tarzında çizimlere dönüştürülmek üzere kırpabilmek ve daha sonraki aşama olan baloncuk yerleştirme aşamasında baloncukların pozisyon ve şekillerini belirleyebilmek amacıyla bir karakter tanıma modeli eğitilmesine karar verilmiştir. Roboflow aracı kullanarak bir object detection projesi oluşturulmuş ve frame verileri eklenmiştir. Frameler bounding box metoduyla seçilmiş ana karakterlerin kafa ve tüm vücudunu alacak şekilde etiketlenmiştir. Toplamda 12 adet sınıf oluşmuştur: *buckbeak, buckbeak-head, draco, draco-head, hagrid, hagrid-head, harry, harry-head, hermonie, hermonie-head, ron, ron-head.*



Şekil 2.2.1: Veri seti etiketleme üzerine örnek

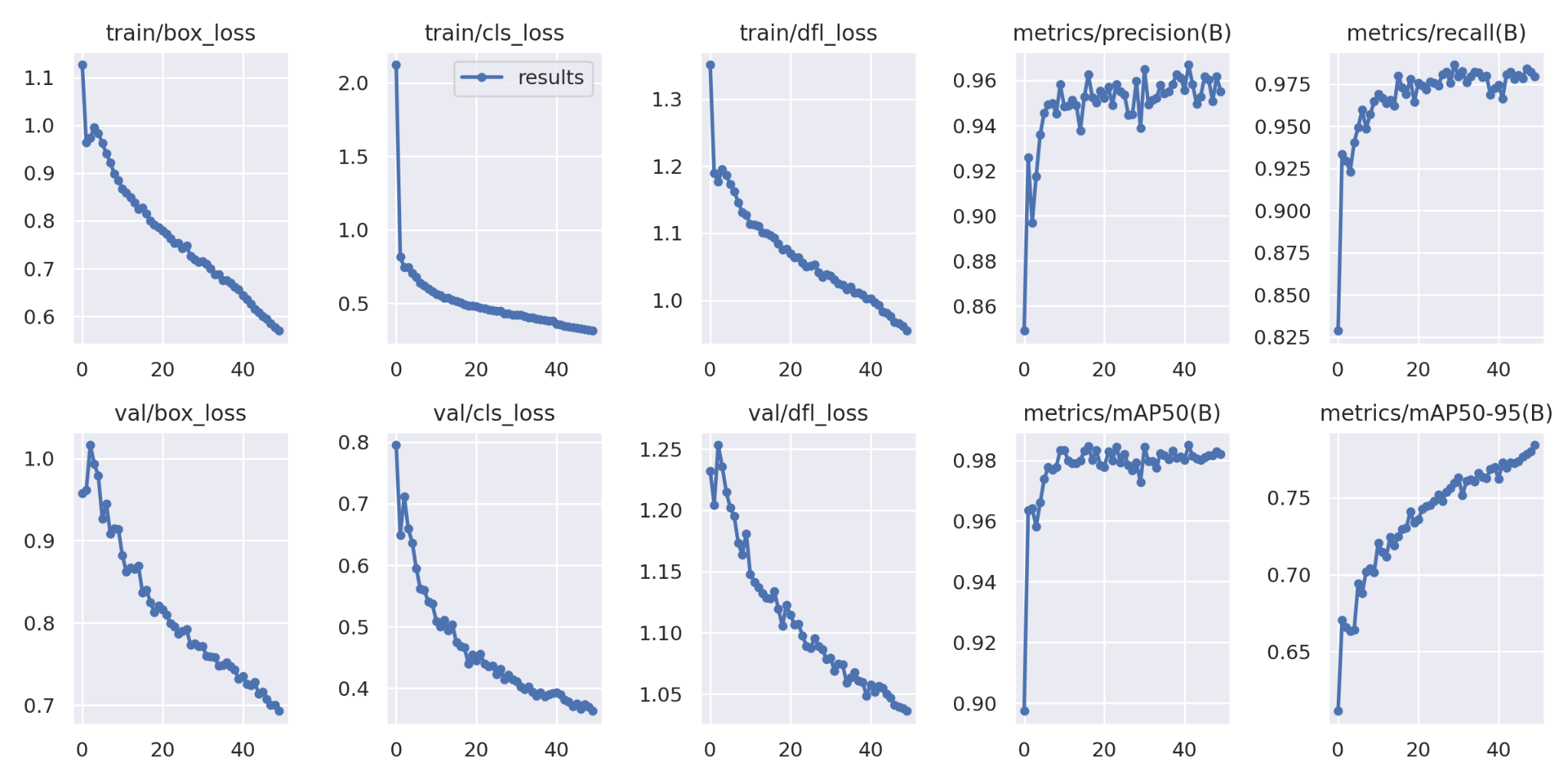
Etiketleme işlemi sonrası etiketli veriler projenin veri setine eklenmiştir. Veri setindeki görüntüler üzerinde ön işleme olarak auto-orient ve resize (640x640) işlemi, veri artırma olarak yatay yansıtma yöntemi ile veri artırma işlemi uygulanıp bir versiyon üretilmiştir. Bu versiyonu indirmek için gerekli Jupyter notebook kodu alınmıştır.



Şekil 2.2.2: Modelin karakterleri detect etmesi üzerine test çıktısı

Eğitim için Google Colab üzerinde bir notebook oluşturulmuştur. Train modunda YOLOv8 modeli kullanılarak veri setinin eğitim işlemine geçilmiştir. *epoch=50*, *batch=16* ve *girdi boyutu (imgsz) 800* *piksel* olarak belirlenmiştir. Eğitilmiş modelin en iyi ağırlıklarının yer aldığı *best.pt* dosyası kullanılarak modelin doğruluğu test edilmiştir. Bu kodda herkese açık şekilde paylaşılmaması gereken API key bilgisi bulunduğu için kaynak kodda zip dosyası üzerinden işlem yapılacak şekilde revize yapıldıktan sonra ipynb uzantılı notebook dosyası ve elde edilen best.pt dosyası GitHub repository’sine eklenmiştir.

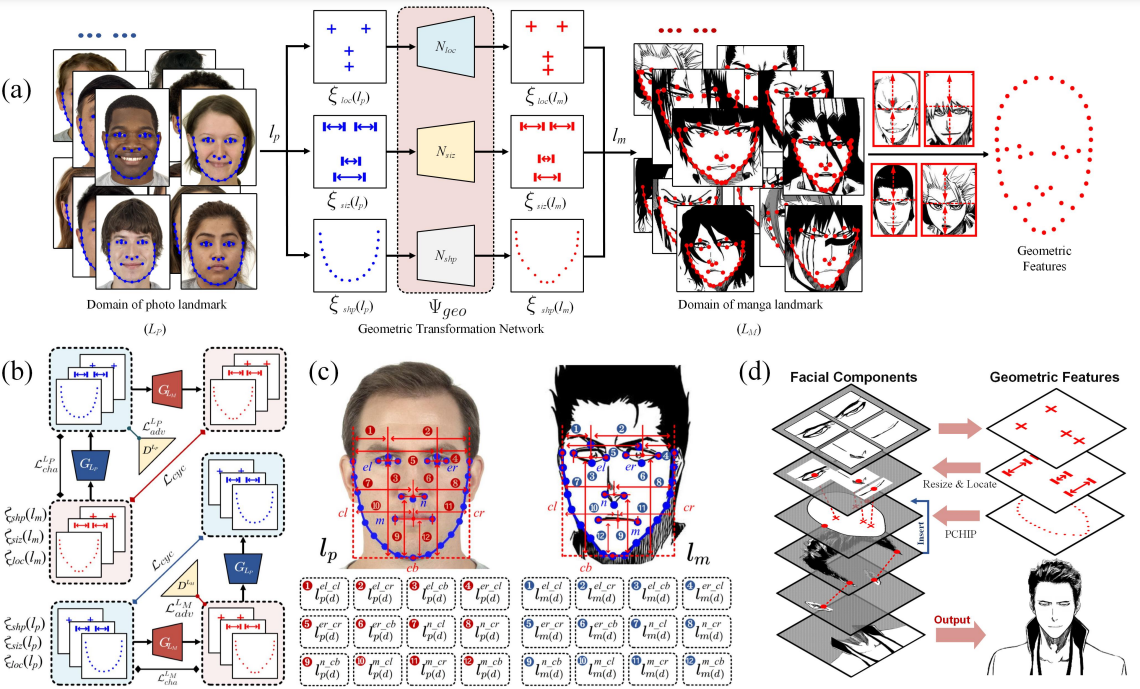
Şekil 2.2.3 ‘teki grafiklerde görüldüğü üzere azalan kayıp değerleri ve yüksek performans metrikleri, modelin başarılı bir şekilde eğitildiğini ve genelleme yeteneğinin iyi olduğunu göstermektedir. Modelin yüksek *hassasiyet (precision)* ve *hatırlama (recall)* oranlarından görüldüğü üzere, doğru nesne tespiti yapabildiğini ve genel performansının yüksek olduğu gözlenmiştir.



Şekil 2.2.3: Karakter tanıma modeline ait kayıp ve performans metrik grafikleri

* + 1. **Manga tarzında çizim**

Çizim aşamasında temel olarak MangaGAN [8] çalışmasından esinlenilmiştir. İlgili çalışmada bulunan ATN (Appearance Transformation Network) mimarisi bu çalışmada farklı tasarlanmıştır. Gerçek insanın göz, burun ve ağız bölgelerini manga tarzında çizime çeviren bir GAN modeli oluşturmak yerine göz, burun ve ağız manga görsellerinden oluşan bir görsel havuzu kullanılması tercih edilmiştir. MangaGAN çalışmasında olduğu gibi, karakterlerin yalnızca yüzlerinin çizdirilmesi ve vücut kısmının görüntü işleme yöntemleriyle manga tarzına çevrilmesi kararlaştırılmıştır.



Şekil 2.2.4: GTN mimarisi [8]

* + - 1. **GTN (Geometric Transformation Network)**

Şekil, konum ve boyut için üç alt GAN oluşturulmuştur: , , . Bu üç GAN kabaca CycleGAN [19] mimarisini izlemektedir. Her bir GAN modelinde iki üretici (generator) ağ ve iki ayrımcı (discriminator) ağ tasarlanmıştır, bu ağlar tam bağlı (fully-connected) sinir ağlarıdır:

* G\_H2M: İnsan domain’inden manga domain’inine dönüşüm yapar.
* G\_M2H: Manga domain’inden insan domain’inine dönüşüm yapar.
* D\_H: Gerçek ve üretilmiş insan değerlerini ayırt eder.
* D\_M: Gerçek ve üretilmiş manga değerlerini ayırt eder.

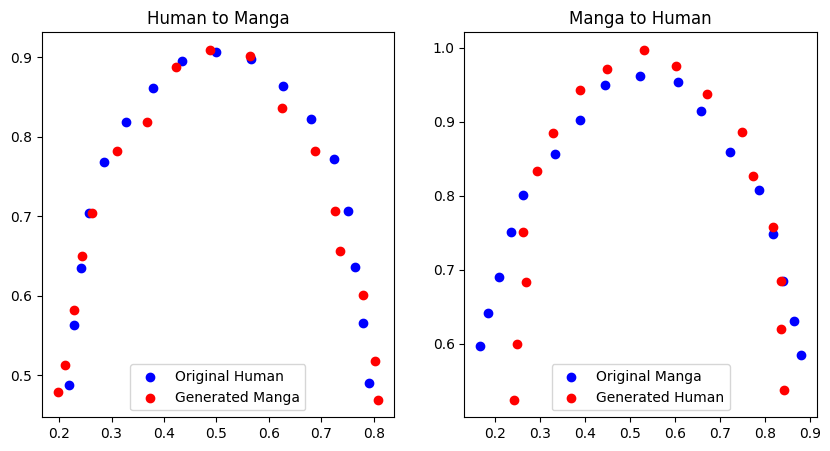
Üretici ve ayrımcı performansını değerlendirmek için Binary Cross-Entropy Loss (BCE) kullanılmıştır. Dönüştürülen landmark'ların orijinal alana doğru şekilde yeniden oluşturulabilmesini sağlamak için L1 Loss kullanılmıştır. Tüm ağlar için öğrenme oranı 0.0002 ve betaları 0.5 ve 0.999 olan Adam optimizatörleri kullanılmıştır. Eğitim süreci 500 epoch boyunca yürütülmüştür.

* + - * 1. **ShapeGAN**

, çenenin şeklini dönüştüren GAN’dır. Amaç, geometrik özellikleri koruyarak iki farklı alan arasında (insan ve manga yüzleri) stili dönüştürmektir. ve veri setlerinde her bir yüze karşılık bir metin dosyası bulunmaktadır. Her bir dosyada 17 adet landmark koordinatlarını içeren 34 değer bulunmaktadır.

Üretici ağlar; 34 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 136, 272 ve 136 nöronlu üç gizli katman ve 34 nöronlu çıkış katmanından oluşmaktadır.

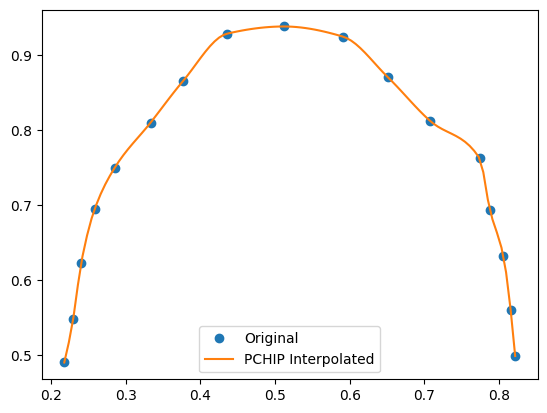
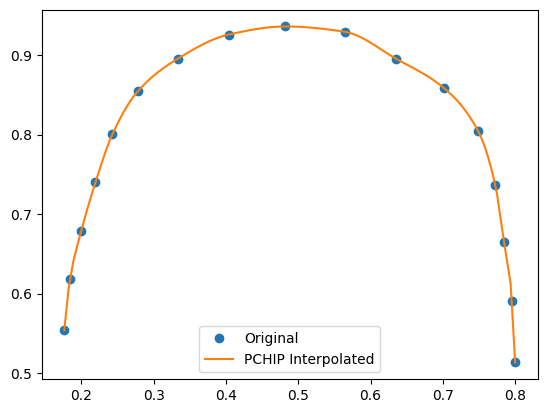
Ayrımcı ağlar; 34 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 136 ve 68 nöronlu iki gizli katman ve Sigmoid aktivasyon fonksiyonu ile takip edilen tek nöronlu çıkış katmanından oluşmaktadır.



Şekil 2.2.5 : Girdi veri setinden alınan verilerin test edilmesi

Veri setleri, eğitim sırasında verimli batch işlemesi sağlamak için batch boyutu 11 olan DataLoader nesnelerine yüklenmiştir. Veri setindeki veriler ile test yapılmış ve üretilen insan ve manga landmark'ları Matplotlib kullanılarak çizilmiştir.

Landmark'lar x-koordinatına göre sıralanmış ve düzgün görselleştirme için Piecewise Cubic Hermite Interpolating Polynomial (PCHIP) yöntemi kullanılarak enterpole edilmiştir.



(a) (b) (c)

Şekil 2.2.6 : shapeGAN’ın sahneden alınan bir kesit ile test edilmesi (a) Gerçek imaj (b) Gerçek imajdan elde edilen landmark koordinatlarından Matplotlib ve PCHIP yöntemi ile çenenin çizdirilmesi (c) shapeGAN’ın G\_H2M modelinin ürettiği manga landmark koordinatlarından Matplotlib ve PCHIP yöntemi ile çenenin çizdirilmesi

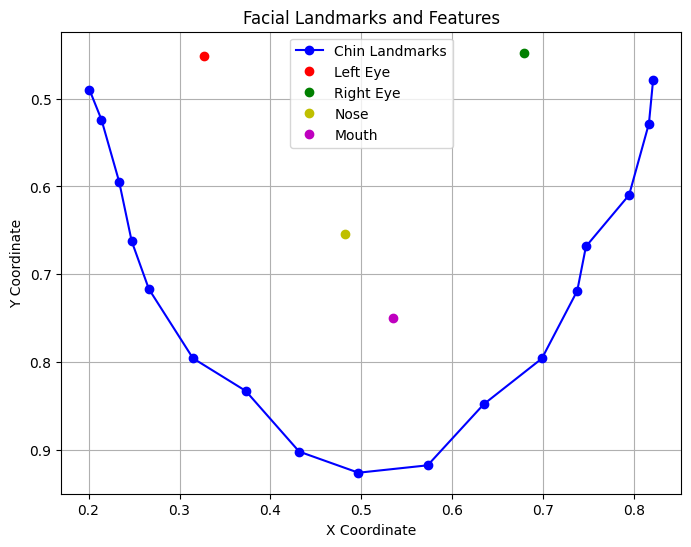
* + - * 1. **locationGAN**

, yüz bileşenlerinin (sol göz, sağ göz, burun ve ağız) çenenin sol uç noktasına, sağ uç noktasına ve dip noktasına olan uzaklıklarının normalize edilmiş x ve y değerlerini dönüştüren GAN’dır. Amaç, bileşenlerin yüzdeki konumlarını dönüştürmektir. ve veri setlerinde her bir yüze karşılık bir metin dosyası bulunmaktadır. Her bir dosyada bileşenleri ifade eden 4 satır ve uzaklıkları ifade eden 3 değer bulunmaktadır. Toplam 12 adet değer bulunmaktadır.

Üretici ağlar; 12 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 48, 96 ve 48 nöronlu üç gizli katman ve 12 nöronlu çıkış katmanından oluşmaktadır.

Ayrımcı ağlar; 12 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 48 ve 24 nöronlu iki gizli katman ve Sigmoid aktivasyon fonksiyonu ile takip edilen tek nöronlu çıkış katmanından oluşmaktadır.

Veri setleri, eğitim sırasında verimli batch işlemesi sağlamak için batch boyutu 5 olan DataLoader nesnelerine yüklenmiştir. Test aşamasında bir görselin landmark bilgileri daha önce oluşturulan shapeGAN modeline verilmiş ve çene landmark’ları ürettirilmiş, uzaklık bilgileri locationGAN modeline verilmiş ve uzaklık bilgileri ürettirilmiş; elde edilen sonuçlar Matplotlib yardımıyla görselleştirilmiştir. Bileşenlerin x koordinatında sol uç noktaya olan uzaklıkları dikkate alınmıştır.



Şekil 2.2.7: shapeGAN ve locationGAN çıktılarının grafikte birleştirilmesi

* + - * 1. **sizeGAN**

, yüz bileşenlerinin (sol göz, sağ göz, burun ve ağız) genişliklerinin normalize edilmiş değerlerini dönüştüren GAN’dır. Amaç, bileşenlerin boyutlarını dönüştürmektir. Yükseklik bilgisi genişlik ile orantılı olacak şekilde sabitlenmiştir, bu yüzden yükseklik bilgisi dönüştürülmemiştir. ve veri setlerinde her bir yüze karşılık bir metin dosyası bulunmaktadır. Her bir dosyada bileşenleri ifade eden 4 satır ve genişlikleri ifade eden 1 değer bulunmaktadır. Toplam 4 adet değer bulunmaktadır.

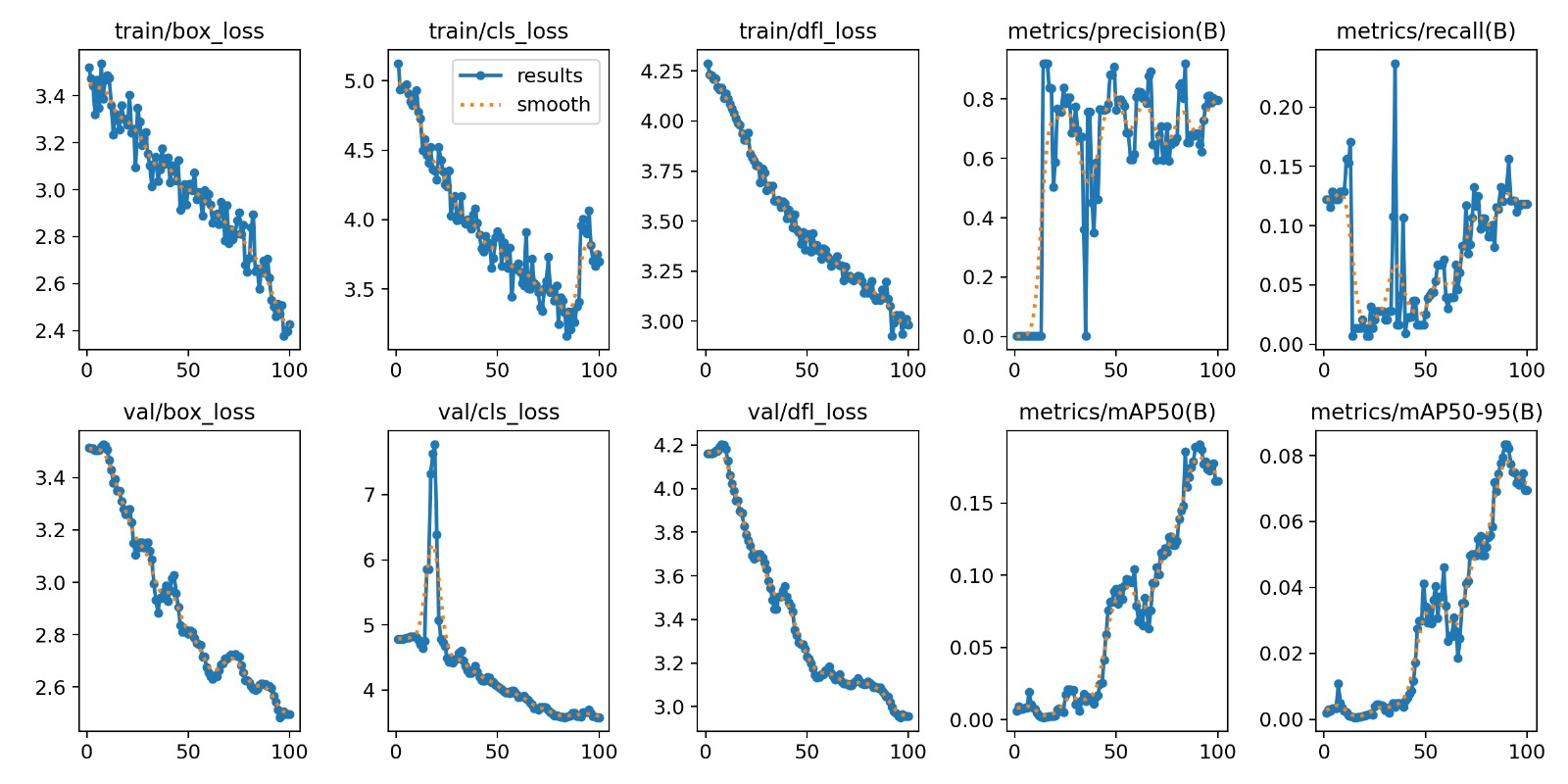
Üretici ağlar; 4 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 48, 96 ve 48 nöronlu üç gizli katman ve 4 nöronlu çıkış katmanından oluşmaktadır.

Ayrımcı ağlar; 4 nöronlu giriş katmanı, her biri ReLU aktivasyon fonksiyonuyla takip edilen 48 ve 24 nöronlu iki gizli katman ve Sigmoid aktivasyon fonksiyonu ile takip edilen tek nöronlu çıkış katmanından oluşmaktadır.

Veri setleri, eğitim sırasında verimli batch işlemesi sağlamak için batch boyutu 5 olan DataLoader nesnelerine yüklenmiştir.

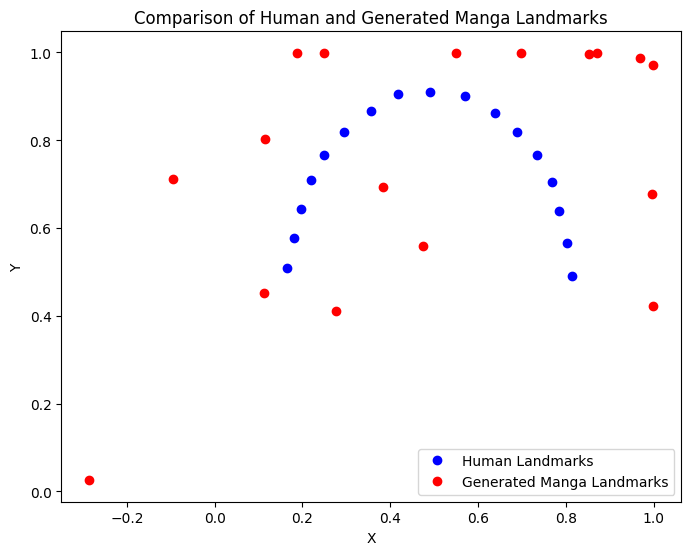
# BULGULAR VE TARTIŞMA

Karakter tanıma modeli için ilk aşamada manuel olarak oluşturulan 100 adet fotoğraftan oluşan veri seti üzerinde çalışma yapılmıştır. Bu model 100 epoch boyunca eğitilmiştir. Şekil 3.1’de görüldüğü üzere, grafikler genel olarak modelin eğitim süreci boyunca performansının arttığını ve kayıplarının azaldığını göstermektedir. Ancak, precision ve recall metriklerindeki dalgalanmalar, modelin bazı nesneleri tespit etmekte zorlandığını göstermektedir. Bundan dolayı veri seti OpenCV kullanılarak elde edilen frameler ile genişletilmiştir ve genişletilmiş veri seti kullanılarak yeni bir model eğitimi gerçekleştirilmiştir. 7838 adet veri kullanılarak elde edilen bulguların 100 adet veriden elde edilen bulgulara göre daha başarılı bulunduğu gözlenmiştir.



Şekil 3.1: İlk geliştirilen karakter tanıma modeline ait çıktılar

Manga tarzı çizdirme adımında ilk aşamada keras kütüphanesi kullanılarak oluşturulmuş, ve veri setleri normalize edilmeden, piksel değerleri ile elde edilmiştir. Bu şekilde yapılan eğitimlerde başarılı bir sonuç elde edilememiştir. Şekil 3.2’de görüldüğü üzere, çeneye benzemeyen ve negatif değerler de üreten bir model oluşmuştur.

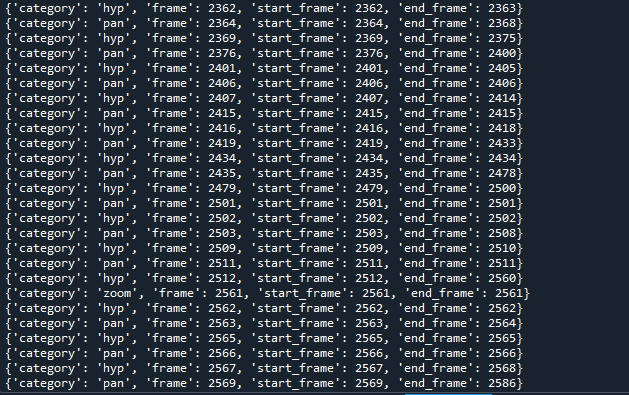


Şekil 3.2: Gerçek piksel değerleri ve keras ile yapılan eğitim sonucu üretilen landmark’lar

CycleGAN mimarisinin, koordinatların x ve y değerlerini dönüştürme yöntemiyle başarılı bir stil transferi yapabildiği görülmüştür. ShapeGAN üzerinde elde edilen başarı sonucunda locationGAN model geliştirmesinde de normalize edilmiş verilerden yararlanılarak hareket edilmiştir.

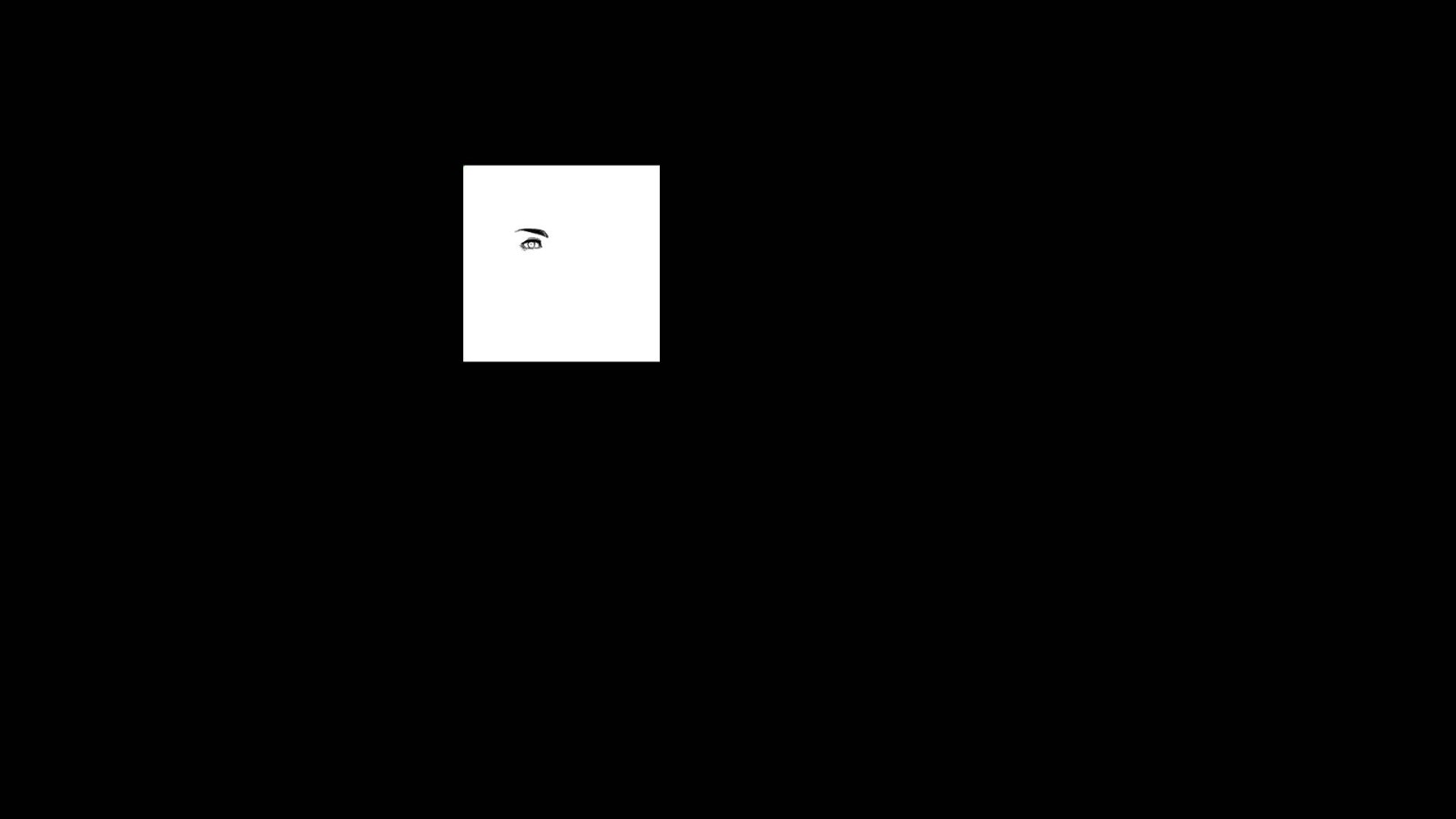
Konuşan kişinin belirlenmesi ve script eşleştirmesi aşamasında Movie2Comics çalışmasında yer alan konuşan ve konuşmayan olarak sınıflandırma işlemi bu çalışmada yapılmamıştır. Karakterleri tanıyacak YOLOv8 pre-trained modelinden yararlanarak bir model oluşturulduktan sonra, scriptte bulunan konuşmacı ismine ait etiketlemeye göre konuşan kişinin frame’de bulunduğu alan tespit edilmiştir.

Optik akış teorisine göre elde edilen hareket değişime göre videodaki belirli bir hareket özelliklerini paylaşan ardışık karelerin oluşturduğu shot segment dizisine ait 6631 adet frame bulunmuş ve bu hareket dizisinde hareketin başlangıç ve bitiş frameleri görüntülenmiştir (Şekil 3.3). Burada optik akış ile ilgili daha hassas bir parametre hesaplaması kullanılarak sonuçlar daha minimal hale getirilebilir.



Şekil 3.3: Belirli hareket özelliklerine ait başlangıç bitiş frameleri

Son aşamada, elde edilen frame’lerden biri, çizim aşamasını test etmek için kullanılmıştır. Frame alınıp karakter tanıma modeli ile karakter tespiti yapıldıktan sonra karakterin kafa bölgesi alınmıştır. Bu bölge GAN veri setlerinde olduğu gibi kare olmak zorunda değildir, bu yüzden elde edilen kafa görseli kırpılarak kare hâle getirilmiştir. Elde edilen kare görüntüdeki yüzün landmark koordinatları dlib kütüphanesi yardımıyla elde edilmiştir. Bu koordinat ve boyut bilgileri shape.txt, location.txt ve size.txt olmak üzere üç dosyaya aktarılmıştır. Buradaki bilgiler , ve ağlarından geçirilerek shape\_result.txt, location\_result.txt ve size\_result.txt dosyaları üretilmiştir. Orijinal görüntüdeki kafa bölgesi ile aynı boyutta bir görüntü oluşturulup bu görüntü üzerine çıktı dosyalarındaki bilgiler yardımıyla hazır sol göz görseli yerleştirilmiştir.



Şekil 3.4: Üretilen shape\_result.txt, location\_result.txt ve size\_result.txt dosyaları kullanılarak çizdirme işlemi sonucu

# SONUÇLAR

Bu projede 2004 yapımı Harry Potter ve Azkaban Tutsağı filminde bulunan “Meet Buckbeak” sahnesindeki karakterlerin Jujutsu Kaisen adlı manga tarzı ile çizdirilmesi ve sonuçların mangalaştırılması amaçlanmıştır. Bu kapsamda key frame seçimi, karakter tanıma modeli, layout seçimi ve manga tarzında çizim aşamaları gerçekleştirilmiştir.

Layout seçimi aşamasında hazır bir layout seçilmiştir. Bunun yerine Content-Aware Video2Comics With Manga-Style Layout [5] çalışmasında olduğu gibi Monte Carlo örnekleme algoritması kullanılarak optimum layout üretilebilir.

Manga tarzı çizdirme aşamasında MangaGAN'dan farklı olarak, ATN oluşturmak yerine göz, burun ve ağız görsellerinden oluşan hazır bir havuz kullanılması tercih edilmiştir. Bunun yerine ATN mimarisi oluşturarak göz, burun ve ağız bölgelerini manga tarzında çizime çeviren GAN'ların kullanılması, gerçek fotoğraflara daha benzer çıktılar almayı sağlayabilir.

Üretilen frame’lerin manga panellerine yerleştirilmesi ve baloncuk yerleştirme aşamaları gerçekleştirilememiştir. Gelecekte bu projede, bu aşamalar için çalışmalar yapılması planlanmaktadır. Frame’lerin panellere yerleştirilmesi aşamasında optimum layout oluşturma kullanılırsa frame’lere uygun paneller oluşmuş olur, böylece daha açıklayıcı manga sayfaları elde edilebilir. Baloncuk yerleştirme aşamasında karakter tanıma modeli de kullanılarak, frame’de mümkün olduğunca karakterin vücudunu kapatmayacak şekilde yerleştirme yapılabilir. Vücut kapanmadan baloncuk yerleştirilemiyorsa kafa önceliklendirilerek kafa bölgesi dışında bulunan bir bölgeye yerleştirilebilir. Baloncuğun konuşmacıya işaret eden uzantısı yine karakter tanıma modeli yardımıyla çizdirilebilir.

Bu projede ortamdaki duygu geçişleri hesaba katılmamıştır. Fakat gelecekte bu projede ek olarak ortamdaki duygu ifadelerini ve yansıma kelimeler çeşitli çizim efektleri kullanılarak oluşturulan mangaya yansıtılabilir.

Projede konuşmaların (script) elde edilmesi aşamasında hazır bir script verisinden yararlanılmıştır. Hali hazırda bulunan bir script dosyasında zaman ve konuşan kişiye ait iki farklı etiket ifadesi yer almaktadır. Script verisini almak için Speech to Text Converter yapısı kullanılarak farklı video türleri için script elde edilmesi otomatikleştirilebilir.

# KAYNAKÇA

[1] “Çekişmeli üretici ağ,” Wikipedia.org, May 29, 2021. <https://tr.wikipedia.org/wiki/%C3%87eki%C5%9Fmeli_%C3%BCretici_a%C4%9F> (Erişim tarihi: 10 Aralık 2023)

[2] W. Hwang, P.-J. Lee, B.-K. Chun, D.-S. Ryu, and H.-G. Cho, “CINEMA COMICS : CARTOON GENERATION FROM VIDEO STREAM,” Jan. 2006, doi: <https://doi.org/10.5220/0001354302990304>.

[3] D.-S. Ryu, S.-H. Park, J. Lee, D. Lee, and H.-G. Cho, “CINETOON: A Semi-automated System for Rendering Black/White Comic Books from Video Streams,” Jul. 2008, doi: <https://doi.org/10.1109/cit.2008.workshops.42>.

[4] M. Wang, R. Hong, X.-T. Yuan, S. Yan, and T.-S. Chua, “Movie2Comics: Towards a Lively Video Content Presentation,” *IEEE Transactions on Multimedia*, vol. 14, no. 3, pp. 858–870, Jun. 2012, doi: <https://doi.org/10.1109/tmm.2012.2187181>.

[5] G. Jing, Y. Hu, Y. Guo, Y. Yu, and W. Wang, “Content-Aware Video2Comics With Manga-Style Layout,” *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2122–2133, Aug. 2015, doi: <https://doi.org/10.1109/tmm.2015.2474263>.

[6] Maciej Pęśko, A. Svystun, P. Andruszkiewicz, P. Rokita, and Tomasz Trzciński, “Comixify: Transform Video Into Comics,” *Fundamenta informaticae*, vol. 168, no. 2–4, pp. 311–333, Sep. 2019, doi: <https://doi.org/10.3233/fi-2019-1834>.

[7] C. Yang, Y. Lai, and Y.-J. Liu, “CartoonGAN: Generative Adversarial Networks for Photo Cartoonization,” *ORCA Online Research @Cardiff (Cardiff University)*, Jun. 2018, doi: https://doi.org/10.1109/cvpr.2018.00986.

[8] H. Su, J. Niu, X. Liu, Q. Li, J. Cui, and J. Wan, “MangaGAN: Unpaired Photo-to-Manga Translation Based on The Methodology of Manga Drawing,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, pp. 2611–2619, May 2021, doi: <https://doi.org/10.1609/aaai.v35i3.16364>.

[9] “Roboflow: Go from Raw Images to a Trained Computer Vision Model in Minutes.,” *roboflow.ai*. <https://roboflow.com/> (Erişim tarihi: 10 Aralık 2023)

[10] “colab.google,” *colab.google*. <https://colab.google/>

[11] G. Jocher, A. Chaurasia, and J. Qiu, Ultralytics YOLOv8. 2023.

[12] “Derin Öğrenme Algoritması (YOLO) Ile Fotoğraflardaki Ya Da Canlı Görüntü Üzerinden Elektrik Sayaçlarının Tespiti.” tr.linkedin.com, [www.linkedin.com/pulse/derin-%C3%B6%C4%9Frenme-algoritmas%C4%B1-yolo-ile-foto%C4%9Fraflardaki-ya-onur-i%CC%87meci/](http://www.linkedin.com/pulse/derin-%C3%B6%C4%9Frenme-algoritmas%C4%B1-yolo-ile-foto%C4%9Fraflardaki-ya-onur-i%CC%87meci/). (Erişim tarihi: 13 Mayıs 2024)

[13] “JoinDatawithme/Humanface\_of\_various\_age\_groups · Datasets at Hugging Face,” huggingface.co. https://huggingface.co/datasets/JoinDatawithme/Humanface\_of\_various\_age\_groups (Erişim tarihi: 13 Mayıs 2024)

[14] N. Alageel, “nawafalageel/Side-Profile-Detection,” *GitHub*, Jun. 23, 2024. https://github.com/nawafalageel/Side-Profile-Detection (Erişim tarihi: 13 Mayıs 2024).

[15] “Jujutsu Kaisen (TV Series 2020– ) - Awards - IMDb,” *www.imdb.com*.

<https://www.imdb.com/title/tt12343534/awards/>

[16] “dlib C++ Library,” *dlib.net*. <http://dlib.net/> (Erişim tarihi: 13 Mayıs 2024)

[17] “i·bug - resources - Facial point annotations,” Ic.ac.uk, 2019. https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/

[18] J. Kim, HO. Chang, J. K, and H. Kim, “Efficient camera motion characterization for mpeg video indexing,” in Proc. IEEE Int. Conf. Multimedia and Expo, 2000.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks,” arXiv.org, 2017. <https://arxiv.org/abs/1703.10593>

# EKLER

* 1. **EK 1: Jujutsu Kaisen mangasına ait diyalog olmayan panel**

****

* 1. **EK 2: Kahramanlık Akademim mangasına ait diyalog olmayan paneller**

****

**6.3. EK3: Mangalarda ses ve duygu efektlerinin kullanımına yönelik örnekler.**

****

1. “Meet Buckbeak | Harry Potter and the Prisoner of the Azkaban”: https://www.youtube.com/watch?v=71xBu\_VHTfY [↑](#footnote-ref-0)