Giriş:

* Proje hedefleri ve amaçları
* Projenin önemi ve kullanım alanları
* Gereksinim Analizi:
* Kullanıcı ihtiyaçlarının belirlenmesi
* Donanım ve yazılım gereksinimleri
* Proje kapsamı ve sınırlamaları

Tasarım:

* Genel proje yapısı ve bileşenler
* Veri akış diyagramları veya iş akışı şemaları
* Kullanıcı arayüzü tasarımı
* Veri tabanı tasarımı (varsa)
* Kullanılan Teknolojiler:
* Kullanılan programlama dili ve çerçeveler
* Kütüphaneler ve dış kaynaklar

Veri Hazırlama:

* Kullanılacak veri setleri
* Veri temizleme ve ön işleme adımları
* OCR Uygulaması:
* Metin çıkarımı için kullanılan yöntemler ve algoritmalar
* Resim işleme adımları
* OCR doğruluk ve performans değerlendirmesi

Kullanıcı Arayüzü:

* Ana ekran ve kullanıcı etkileşimi
* Butonlar, giriş alanları, sonuç gösterimi vb.
* Test ve Doğrulama:
* Birim testleri ve test senaryoları
* Performans ölçümleri
* Hata ayıklama ve iyileştirme adımları
* Dağıtım ve Yayınlama:
* Uygulamanın dağıtımı için kullanılan yöntemler (ör. derleme, paketleme)
* Desteklenen platformlar ve işletim sistemleri

Kullanım Kılavuzu:

* Kurulum talimatları
* Uygulama kullanımı ve özellikleri
* İlerleme Planı:
* Projenin mevcut durumu ve ilerleme takvimi
* Yaklaşan özellikler veya güncellemeler

Kaynaklar:

* Kullanılan literatür, referanslar ve diğer kaynaklar
* İletişim bilgileri veya projenin yer aldığı bir web sitesi (varsa)

**1.Giriş**

Bu proje, Optik Karakter Tanıma (OCR) teknolojisinin kullanıldığı bir metin çıkarım uygulamasını geliştirmeyi amaçlamaktadır. Bu uygulama, görüntülerdeki yazıları otomatik olarak tanıyabilen bir teknolojiyi kullanarak metin çıkarımı yapmayı hedeflemektedir.

Günümüzde, yazılı metinlere erişim önemli bir gereksinim haline gelmiştir. Ancak bazen yazılı belgelere erişmek zor olabilir veya elde yazılı bir kopya bulunmayabilir. Bu noktada, OCR teknolojisi devreye girer. Bu projenin amacı, görüntülerdeki metni tanıyarak, yazılı belgeleri dijital ortama aktarmak ve üzerinde çeşitli işlemler yapabilmek için kullanıcılara bir araç sunmaktır.

Bu proje, birçok alanda kullanılabilen geniş bir kullanım potansiyeline sahiptir. Örneğin, belge tarama süreçlerinde kullanılarak zaman ve emek tasarrufu sağlanabilir. Ayrıca, otomasyon sistemlerinde veri girişi yaparken kullanılabilir ve bu sayede insan hatasını minimize edebilir. Dijital arşivleme, kitap ve gazete tarama, anket analizi gibi birçok alanda da kullanılabilir.

Bu projenin temel amacı, kullanıcıların günlük işlerinde metin çıkarımı yaparken daha kolay ve verimli bir deneyim yaşamalarını sağlamaktır. Kullanıcılar, OCR teknolojisinin sunduğu avantajları kullanarak zaman tasarrufu yapabilir, daha hızlı veri erişimi elde edebilir ve iş akışlarını optimize edebilirler.

Bu projenin başarılı olması için, OCR teknolojisini etkili bir şekilde uygulayabilmek, kullanıcı dostu bir arayüz sunmak ve güvenilir sonuçlar elde etmek önemlidir. Bu dokümanda, projenin tasarımı, kullanılan teknolojiler, kullanım senaryoları ve geliştirme süreci gibi konulara değinilecektir. Ayrıca, projenin ilerleyen aşamalarında yapılması planlanan geliştirmeler ve projenin potansiyel kullanım alanları da ele alınacaktır. Youtube linki.

**2.Tasarım**

2.1 Genel Proje Yapısı ve Bileşenler:

Projenin genel yapısı, kullanıcı arayüzü ve metin çıkarımı işlemlerini gerçekleştiren bileşenlerden oluşmaktadır. "new2.py" dosyası içindeki kodlar bu bileşenlerin işlevselliğini sağlamaktadır. Aşağıda projenin temel bileşenlerini açıklayan bir genel proje yapısı sunulmuştur:

Kullanıcı Arayüzü: Projenin kullanıcı arayüzü, Tkinter kütüphanesi kullanılarak oluşturulmuştur. Kullanıcı, görüntü seçme, yapıştırma ve metin çıkarımı gibi işlemleri bu arayüz üzerinden gerçekleştirebilir.

Görüntü İşleme: Proje, PIL (Python Imaging Library) kütüphanesini kullanarak görüntü işleme işlemlerini gerçekleştirir. Seçilen görüntüyü açar, metin çıkarımı için gerekli işlemleri uygular ve sonuçları kullanıcıya gösterir.

Metin Çıkarımı: Metin çıkarımı işlemi, pytesseract kütüphanesi kullanılarak gerçekleştirilir. Bu kütüphane, görüntülerdeki metni tanıma yeteneğine sahiptir. Proje, pytesseract kütüphanesini kullanarak seçilen görüntüdeki metni çıkarır ve kullanıcıya sunar.

Veri Akış Diyagramları veya İş Akışı Şemaları:

Proje için veri akış diyagramları veya iş akışı şemaları, projenin karmaşıklığına ve detay seviyesine bağlı olarak oluşturulabilir. Temel olarak, kullanıcının yapacağı işlemleri ve bu işlemlerin nasıl gerçekleştirileceğini gösteren bir akış şeması oluşturulabilir. Örneğin, kullanıcının görüntü seçme işleminden başlayarak metin çıkarımı işlemine kadar olan adımları gösteren bir akış şeması oluşturulabilir.

2.2 Kullanıcı Arayüzü Tasarımı:

Projenin kullanıcı arayüzü, "new2.py" dosyasında kodlanmıştır. Arayüz, Tkinter kütüphanesinin sağladığı widget'lar ve düzenleme özellikleri kullanılarak oluşturulmuştur. Arayüz, başlık etiketi, seçim düğmesi, yapıştırma etiketi, sonuç etiketi gibi bileşenleri içermektedir. Kullanıcı, arayüz üzerinden görüntü seçebilir, metin çıkarımı yapabilir ve sonuçları görebilir.

2.3 Veri Tabanı Tasarımı (varsa):

Bu projede veri tabanı kullanımı öngörülmemektedir. Veri tabanı kullanımının gerekliliğine bağlı olarak, projenin ilerleyen aşamalarında veri tabanı tasarımı ve entegrasyonu gerçekleştirilebilir.

2.4 Kullanılan Teknolojiler:

Kullanılan Programlama Dili ve Çerçeveler: Proje, Python programlama dilinde geliştirilmiştir. Python, kullanıcı dostu ve geniş bir kütüphane ekosistemiyle birlikte gelir, bu da projenin hızlı ve etkili bir şekilde geliştirilmesini sağlar. Tkinter, kullanıcı arayüzü oluşturmak için kullanılan bir Python kütüphanesidir.

Kullanılan Kütüphaneler ve Dış Kaynaklar: Proje, PIL (Python Imaging Library) ve pytesseract gibi kütüphaneleri kullanmaktadır. PIL, görüntü işleme işlemlerini gerçekleştirmek için kullanılan popüler bir Python kütüphanesidir. pytesseract, OCR (Optik Karakter Tanıma) işlemlerini gerçekleştirmek için kullanılan bir Python kütüphanesidir. Bu kütüphaneler, projenin temel işlevselliğini sağlamak için kullanılmaktadır.

**3. Kullanım Kılavuzu**

3.0. Kurulum Talimatları:

3.1. Python Kurulumu: Proje, Python programlama dili kullanılarak geliştirildiği için ilk adım olarak Python'un bilgisayarınıza yüklü olması gerekmektedir. Python'un en son sürümünü resmi Python web sitesinden indirip kurabilirsiniz.

3.2. Gerekli Kütüphanelerin Kurulumu: Proje, PIL (Python Imaging Library) ve pytesseract gibi kütüphaneleri kullanmaktadır. Bu kütüphaneleri yüklemek için aşağıdaki adımları takip edebilirsiniz:

- PIL Kütüphanesi: Komut istemini açın ve aşağıdaki komutu girin:

Terminal: pip install pillow

- pytesseract Kütüphanesi: Komut istemini açın ve aşağıdaki komutu girin:

Terminal: pip install pytesseract

3.3. Projenin İndirilmesi: Proje dosyalarını indirmek için projenin bulunduğu depoya gidin ve "Clone" veya "Download" seçeneğini kullanarak dosyaları bilgisayarınıza indirin.

Github link:

3.4. Projenin Çalıştırılması: İndirdiğiniz proje dosyalarını bir dizine çıkartın. Ardından, komut istemini açın ve proje dizinine gidin. Aşağıdaki komutu girerek projeyi çalıştırabilirsiniz:

Terminal: python new2.py

3.5. Uygulama Kullanımı ve Özellikleri:

Bu uygulama, görüntülerden metin çıkarmak için kullanılan bir araçtır. Kullanıcı, arayüz üzerinden iki farklı yöntemle metin çıkarımı yapabilir:

3.6. Görüntü Seçme: "Select Picture" düğmesine tıklayarak bir görüntü seçebilirsiniz. Seçtiğiniz görüntü açılacak ve metin çıkarımı işlemi gerçekleştirilecektir. Elde edilen metin, uygulama arayüzünde görüntülenecektir.

3.7. Ekran Görüntüsü Yapıştırma: Uygulama arayüzünde "Press Ctrl+V to paste a screenshot" yazan bir bölüm bulunmaktadır. Bu bölüme, ekran görüntüsü yapıştırabilirsiniz. Yapıştırdığınız ekran görüntüsü üzerinde metin çıkarımı işlemi gerçekleştirilecek ve elde edilen metin, uygulama arayüzünde görüntülenecektir.

3.8. İlerleme Planı:

Uygulama açık kaynak kodlu olarak geliştirilecek olup, ilerleyen süreçte çevrim içi hizmet sunabilme yeteneğine sahip hale getirilmesi planlanmaktadır. Şu an itibarıyla, 06-2023 tarihi itibarıyla sadece Windows platformunda kullanılabilir durumdadır. Projenin tek geliştiricisi ben olduğum için güncellemeler sık ve düzenli bir şekilde gerçekleştirilemeyebilir. Ancak, ilerleyen güncellemelerde öncelikli olarak uygulamanın sunucu tarafında hizmet verebilme yeteneği yeniden kodlanacak ve mikro hizmetlerle yeni özellikler kazandırılacaktır. Ayrıca, yeni bir kullanıcı arayüzü Flutter kullanılarak tüm platformlar için oluşturulacaktır. Bu sayede uygulamanın kullanılabilirliği artacak ve kullanıcılar farklı platformlarda sorunsuz bir deneyim yaşayabilecektir.

3.9. Projenin Mevcut Durumu ve İlerleme Takvimi:

- Şu anda projenin temel işlevselliği tamamlanmış durumdadır. Kullanıcı, görüntü seçme ve ekran görüntüsü yapıştırma yöntemleriyle metin çıkarımı yapabilir. Elde edilen metin uygulama arayüzünde görüntülenebilir.

- İlerleme takvimi, projenin geliştirme hızına ve ek özelliklerin entegrasyonuna bağlı olarak değişebilir. Ancak, mevcut plana göre yakın zamanda aşağıdaki özelliklerin eklenmesi planlanmaktadır:

* Dil Seçeneği: Kullanıcıların metin çıkarımı yaparken kullanacakları dili seçebilecekleri bir özellik eklenmesi planlanmaktadır. Bu sayede kullanıcılar, çeşitli dillerdeki metinleri çıkarabilirler.
* Metin Düzenleme: Kullanıcıların elde ettikleri metinleri düzenleyebilecekleri ve gerekli düzeltmeleri yapabilecekleri bir metin düzenleme özelliği eklenmesi planlanmaktadır.
* Dosya Kaydetme: Kullanıcıların metin çıkarımı sonucunu bir dosyaya kaydetmelerini sağlayacak bir özellik eklenmesi planlanmaktadır. Bu sayede elde edilen metinleri daha sonra kullanmak veya paylaşmak için saklayabilirler.
* Çoklu Görsel Okuma: Projeye çoklu görsel okuma özelliğini eklemek planlanmaktadır. Bu özellik sayesinde kullanıcılar birden fazla görüntüyü aynı anda seçip metin çıkarımı işlemini gerçekleştirebileceklerdir. Bu, kullanıcıların zaman kazanmasını sağlayacak ve daha verimli bir kullanım deneyimi sunacaktır.
* Flutter ile Web ve Mobil Arayüz: Projenin kullanıcı arayüzünü geliştirmek amacıyla Flutter kullanarak hem web hem de mobil platformlar için uyumlu bir arayüz sağlamayı planlanmaktadır. Bu sayede kullanıcılar, projeye web tarayıcılarından veya mobil cihazlarından erişebilecek ve metin çıkarımı işlemlerini kolaylıkla gerçekleştirebileceklerdir.

Bu özelliklerin eklenmesiyle birlikte uygulamanın kullanılabilirliği ve işlevselliği artacak ve kullanıcıların daha iyi bir deneyim yaşaması sağlanacaktır. İlerleme takvimi, bu özelliklerin geliştirilme ve entegrasyon süreçlerine bağlı olarak güncellenecektir.

**4. Kaynaklar:**

4.0. Kullanılan literatür, referanslar ve diğer kaynaklar:

* Pillow Resmi Belgeleri: <https://pillow.readthedocs.io/en/stable/> - Pillow kütüphanesi, görüntü işleme işlemleri için kullanılan resmi belgelere erişim sağlar.
* Pytesseract GitHub Deposu: <https://github.com/madmaze/pytesseract> - Pytesseract, OCR (Optical Character Recognition) işlemleri için kullanılan bir Python kütüphanesidir. GitHub deposu, kaynak koduna ve detaylı kullanım örneklerine erişim sağlar.