

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Image Captioning (Görüntü Altyazılama)/Second School

**PROJE YAZARI**

Yiğit Ali KURT

**DANIŞMAN**

Prof. Dr. SERHAT ÖZEKES

**İL, TEZ YILI**

İstanbul, 2024



**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

Image Captioning (Görüntü Altyazılama)/Second School

**PROJE YAZARI**

Yiğit Ali KURT

O170420025

**DANIŞMAN**

Prof. Dr. SERHAT ÖZEKES

**İL, TEZ YILI**

İstanbul, 2024

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencisi Yiğit Ali KURT ‘un “Second School” başlıklı bitirme projesi çalışması, 03/06/2024 tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Prof. Dr. Adı SOYADI (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

TABLO LİSTESİ…………………………………………………………………………..iv

ÖZET………………………………………………………………………………………..v

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

1.1.1. Bitirme Projesinin Amacı…………………………………………………….1

1.1.2. Literatür Özeti……………………………...………………………………...2

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

1.2.1. Kullanılan ana kütüphaneler..…………………………………..…………….3

1.2.2. Flickr8k……………………..………………………..………………………4

1.2.3. InceptionV3..………………..………………………..………………………4

1.2.4. LSTM………………………..………………………..………………………5

1.2.5. Modelin eğitimi.……………..………………………..………………………6

1.2.6. Eğitilen modelin kullanımı…..………………………..………………………7

BÖLÜM 3. BULGULAR VE TARTIŞMA……………………………………………..…..8

1.3.1. Durum…………………........…………………………………..…………….8

1.3.1. Ne yapılabilir………………..…………………………………..…………….9

BÖLÜM 4. SONUÇLAR……………………………………………………………….....10

KAYNAKLAR…………………………………………………………………………….11

EKLER…………………………………………………………………………………….14

ÖZGEÇMİŞ

**SEMBOLLER/SYMBOLS**

**Kd :** Türev kontrolör katsayısı

**Ki :** İntegral kontrolör katsayısı

**Kp** : Oransal kontrolör katsayısı

**N :** Aktif kural sayısı

**S :** Kanal kesit alanı (m2)

**t :** Zaman (s)

**T :** Sıcaklık (°C)

**KISALTMALAR/ABBREVIATIONS**

**CNN:** Convolutional Neural Network

**RNN:** Recurrent Neural Network

**LSTM:** Long Short Term Memory

**GUI**: Graphical User Interface

**ŞEKİL LİSTESİ**

Sayfa

Şekil 1.1. Küçük çocuklar için bazı eğitim uygulamaları ………………..…………………7

Şekil 1.2. İmage Captioning Modeli Diyagramı …………………………………………...13

Şekil 1.3. InceptionV3 Modeli Diyagramı…….…………………………………………...15

Şekil 1.4. Uygulamanın ana ekranı ve cevap ekranı..……………………………………...17

**TABLO LİSTESİ**

Sayfa

**ÖZET**

"Second School", küçük çocukların okuma ve yazma becerilerini geliştirmek amacıyla tasarlanmış görsel tabanlı bir öğrenme uygulamasıdır. Bu uygulama, çocukların günlük yaşamlarından veya internetten indirdikleri görselleri kullanarak etkileşimli bir öğrenme deneyimi yaşamalarını sağlar. Görseller, çocukların ilgi ve merakını çekmek için kullanıcı tarafından seçilir ve içerdikleri nesnelerin veya kavramların tanınması ve anlamlandırılması için proje bir araç olarak kullanılır.

Projenin önemli bir özelliği, görsel ve metinsel içeriği bir araya getirerek çocuklara farklı öğrenme deneyimleri sunmasıdır. Görseller, çocukların görsel okuryazarlıklarını geliştirmelerine yardımcı olurken, ilişkilendirilen metinler ise okuma, yazma ve dil becerilerini geliştirmelerine katkı sağlar. Bu sayede, çocuklar görsel ve metinsel bilgiler arasında bağlantı kurmayı öğrenir ve çoklu zeka alanlarında kendilerini geliştirme fırsatı bulurlar.

Platform aynı zamanda, çocukların öğrenme sürecine aktif olarak katılımını teşvik etmek için interaktif öğrenme aktiviteleri ve oyunlara da alışmalarına yardımcı olur. Bu aktiviteler, çocukların öğrenmeyi daha eğlenceli hale getirir ve öğrenme motivasyonlarını artırır.

"Second School", kullanıcı dostu arayüzü sayesinde çocukların kolayca erişebilmesini ve kullanabilmesini sağlar. Ayrıca, öğrenme sürecini desteklemek için öğretmenler ve ebeveynler için kaynaklar ve rehberlik materyalleri de sunar.

Sonuç olarak, "Second School" projesi, çocukların okuma ve yazma becerilerini geliştirmek için yenilikçi bir yaklaşım sunar ve onların eğitim yolculuğunun başlangıcında sanki ikinci bir okulmuşçasına önemli bir rol oynar.

**ABSTRACT**

"Second School" is a visual-based learning platform designed to improve the reading and writing skills of young children. This platform allows children to have an interactive learning experience using images from their daily lives or downloaded from the internet. Visuals are chosen by the child to attract their attention and curiosity and are used as tools to recognize and make sense of the objects or concepts they contain.

An important feature of the platform is that it offers children different learning experiences by combining visual and textual content. While visuals help children develop their visual literacy, associated texts contribute to their development of reading, writing and language skills. In this way, children learn to make connections between visual and textual information and have the opportunity to develop themselves in multiple intelligence areas.

The platform also helps familiarize children with interactive learning activities and games to encourage active participation in the learning process. These activities make children's learning more fun and increase their motivation to learn.

"Second School" allows children to easily access and use it thanks to its user-friendly interface. It also offers resources and guidance materials for teachers and parents to support the learning process.

As a result, the "Second School" project offers an innovative approach to improving children's reading and writing skills and plays an important role at the beginning of their educational journey, as if it were a second school.

1. **GİRİŞ**
   1. **Bölüm 1 – Giriş**

Günümüzde teknoloji tüm hızıyla gelişmeye ve ilerlemeye devam etmekte ancak eğitim sisteminde bunun etkisi özellikle de pandemi dönemi ne kadar çok minimaldi. Ve ancak bundan sonra eğitimde dijitalleşme yeni yeni yaygınlaşmakta (Okullarda uzaktan eğitim, videolu eğitim vb. pandemi döneminde aşırı düzeyde yaygınlaşmış olsa da pandeminin sonlanmasından itibaren zorunluluğun kalkması ile beraber çoğu ilkokul, ortaokul ve lisede tamamen, üniversitelerde ise büyük oranda kaldırılmıştır. Ancak pandemi ebeveynlerin ve öğrencilerin zorunluluktan öğrendikleri eğitim uygulamalarının yararlarını fark etmeleri ile eğitim kültürünü kalıcı olarak değiştirmiştir.) ve bunun öncüsü olanlar ise bu alan üzerine geliştirilmiş olan uygulamalardır [1].

kırpıntı çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 1.1.** Küçük çocuklar için bazı eğitim uygulamaları

* + 1. **Amaç**

Ancak şuan bile bu alanda bulunan az sayıda uygulamaların çoğu öğrenciler için ya okuma yazmanın çok daha önceki dönemleri yada okuma yazmadan sonraki dönemleri temel almakta. Direk olarak okuma yazma dönemini temel alan uygulamalar ise hazır görseller ve metin blokları kullanmakta. Bu projenin amacı ise okuma yazma öğrenmekte olan küçük çocukların hazır metinler yada görsellerdense kendilerinin buldukları görselleri kullanarak daha etkileşimli ve ilgi çekici biçimde okuma yazma öğrenmelerine yardımcı olmaktır. Bunun dışında bu proje temel olarak küçük çocukları hedef alsa da yetişkinlerinde yabancı dil öğrenmesine kelimeleri ve cümleleri öğrenirken akıllarında önce durumu sonra kendi dillerinde nasıl belirteceklerini daha da sonra ise yabancı dilde nasıl belirteceklerini düşünmesi halini, belirteceklerini direk olarak yabancı dile bağlayarak yardımcı olmaktadır.

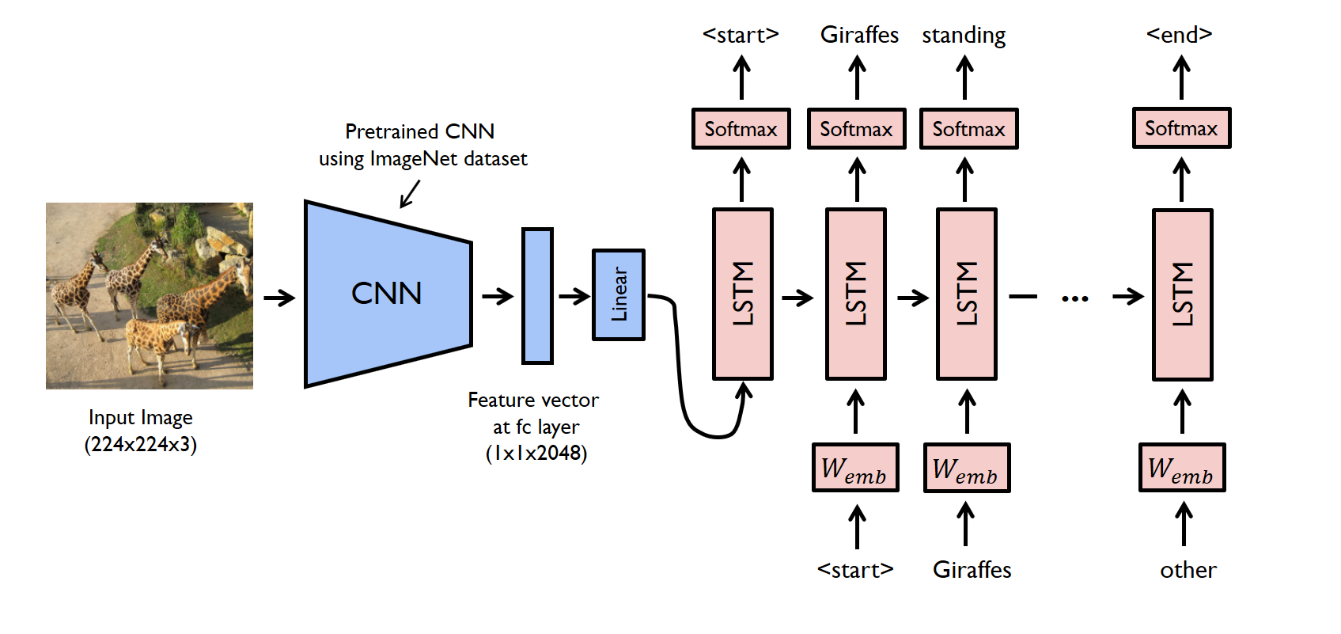
* + 1. **Literatür Özeti**

Yapılan araştırmalar sonucunda bu projenin temelini oluşturması planlanan Image Captioning ile ilgili birçok proje ve kaynak bulundu.

Eğer bulunan projeleri github platformu ile sınırlarsak bulunan image captioning projelerine 49 proje örnek verebiliriz. Bunlardan duruma uygun olan bazılarına ise HeliosX7 tarafından yapılan “image-captioning-app” (2020) [2], purveshpatel511 tarafından yapılan “imageCaptioning” (2020) [3], angeligareta tarafından yapılan “image-captioning” (2021) [4], nithintata tarafından yapılan “image-caption-generator-using-deep-learning” (2020) [5], yada Sajid030 tarafından yapılan “image-caption-generator” ı (2023) [6] örnek verebiliriz. Bunun dışında aktif olarak kullanılan çocuk eğitim uygulamalarını araştırdığımızda ise çok daha az sayıda da olsa ABCmouse [7], Khan Academy Kids [8] yada lingokids [9] gibi örnekler bulundu. Ancak bulunan çocuk eğitim uygulamasında bu tür bir özelliğin olduğuna dair kanıt yada belirti bulunamadı.

* 1. **Bölüm 2 – Materyal ve Yöntem**

Bu proje temel olarak “Image Captioning ” üzerine kurulu bir projedir image captioning ise daha spesifik olarak, bir görüntüden ilgili bilgilerin çıkarılması ve bu görüntünün içeriğini açıklamak için uygun açıklamanın oluşturulması görevidir.



**Şekil 1.2.** İmage Captioning Modeli Diyagramı

Daha basit bir biçimde açıklamak gerekirse image captioning işleminde ilk önce görüntü alınır sonra gelen görüntünün özellikleri bir Evrişimsel Sinir Ağı (Convolutional Neural Network, Bundan sonra CNN olarak bahsedilecektir) modeli ile çıkartılır. Ve çıkartılan özelliklerden bir Tekrarlayan Sinir Ağı (Recurrent Neural Network, bundan sonra RNN olarak bahsedilecektir.) modeli ile mantıklı bir cümle oluşturulur.

* + 1. **Kullanılan ana kütüphaneler:**
       1. **os:** Dosya ve dizin işlemleri için kullanılmıştır.
       2. **pickle:** Nesnelerin serileştirilmesi ve seriden ayrıştırılması için kullanılmıştır.
       3. **numpy (np):** Sayısal hesaplamalar için kullanılmıştır.
       4. **tkinter (tk):** Grafiksel Kullanıcı Arayüzü ( Graphical User Interface Bundan sonra GUI olarak bahsedilecektir ) oluşturmak için kullanılmıştır.
       5. **PIL (Python Imaging Library):** Görüntü işleme işlemleri için (yeniden boyutlandırma vb.) kullanılmıştır.
       6. **tensorflow (tf):** Derin öğrenme modeli oluşturmak ve eğitmek için kullanılmıştır.
    2. **Flickr8k:**

Bu projenin ise model eğitiminde Flickr8 veri seti kullanılmaktadır Flickr8k data setinin kullanılma sebebi ise bu alandaki hazır veri setleri arasında minimal hata payı ile çalışan en küçük veri seti olmasıdır. Bu alanda kullanılan veri setlerine örnek vermemiz gerekirse yaklaşık 32000 görüntü ve 160000 açıklamadan oluşan Flickr30k’yı yaklaşık 3300000000 görüntü ve açıklamadan oluşan Google Conceptual Captions’ i yaklaşık 120000 görüntü ve 600000 açılamadan oluşan MS COCO’ yu yada Hacettepe Üniversitesinin Flickr8k veri setini çevirerek ürettiği 8091 görsel ve 16182 açıklama içeren ve Türkçe dilinde olan TasvirEt’ i örnek verebiliriz. Ama bu projede kullanılmış olan veri seti olan Flickr8k tam olarak 8091 görsel ve her görsel başına 5 açıklama ile toplam 40455 açıklama içermektedir.

* + 1. **InceptionV3:**

Bu projede görselden özellik çıkarmak için InceptionV3 modeli kullanılmaktadır. InceptionV3, derin öğrenme alanında kullanılan bir görüntü sınıflandırma modelidir. Google tarafından geliştirilmiştir ve genellikle büyük veri setleri üzerinde eğitilir(Bu projede orta-küçük bir veri seti üzerinde kullanıldı). Amacı, verilen bir görüntünün içeriğini doğru bir şekilde tanımlamaktır. Örneğin, bir görüntünün içinde bulunan nesneleri sınıflandırmak veya görüntünün hangi kategoriye ait olduğunu belirlemek gibi görevlerde kullanılabilir. InceptionV3 modeli, karmaşık bir evrişimli sinir ağı (CNN) mimarisine sahiptir. Bu ağ, görüntülerdeki çeşitli özellikleri tanımak için birçok katmandan oluşur. İlk katmanlar, basit özellikleri öğrenmek için kullanılırken, daha derin katmanlar, daha karmaşık ve soyut özellikleri öğrenmek için kullanılır. Model, veri setindeki binlerce görüntüden öğrenilir. Bu görüntüler, belirli bir etiket veya kategori ile etiketlenmiştir. Model eğitimi sırasında, ağ, bu görüntülerden özellikler çıkarır ve bu özelliklerle etiketler arasında ilişkiyi öğrenir. Daha sonra, yeni bir görüntü verildiğinde, ağ bu öğrendiği ilişkileri kullanarak görüntünün içeriğini tahmin eder. InceptionV3 modeli, önceden eğitilmiş ağırlıklarla birlikte sunulabilir. Bu, başka bir görev için bu modeli kullanmak istediğinizde, önceki görevlerde başarıyla öğrenilmiş olan özelliklerin yeniden kullanılmasını sağlar. Bu, transfer öğrenme olarak bilinir ve modelin yeni görevlerde daha hızlı ve daha etkili bir şekilde öğrenmesini sağlar. Sonuç olarak, InceptionV3, görüntü sınıflandırma ve diğer görsel işleme görevlerinde kullanılan bir derin öğrenme modelidir. Model, görüntülerdeki özellikleri tanımak için karmaşık bir sinir ağı kullanır ve önceden eğitilmiş ağırlıklarla birlikte sunulur, böylece farklı görevlerde hızlı ve etkili bir şekilde kullanılabilir. Projede bu modelin kullanılma sebebi ise sahip olduğu büyük kullanıcı grubu nedeni ile ilgili kaynakları bulmanın kolaylığı ve sahip olduğu yüksek performanstır.

ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 1.3.** InceptionV3

* + 1. **Uzun kısa süreli bellek:**

Bu projede çıkarılan özelliklerden mantıklı cümleler oluşturma amacıyla kullanılan RNN modeli ise Uzun Kısa Süreli Bellek’dir (Long Short Term Memory, Bundan sonra LSTM olarak bahsedilecektir.). LSTM, özellikle zaman serisi verileri gibi sıralı verilerle çalışırken kullanılır. LSTM'nin diğer geleneksel RNN modellerine göre düşünülürse temel amacı, geleneksel RNN modellerin yaşadığı "uzun vadeli bağımlılık problemini" çözmektir (Projedeki asıl amacı ise verilen özelliklerin oluşturduğu kelime öbeğinde mantıklı bir cümle oluşturmaktır.). Bu problem, geleneksel RNN modellerin uzun bir zaman aralığında önemli bilgileri hatırlayamamasından kaynaklanır. Örneğin, bir cümledeki ilk kelimenin sonunda bulunan önemli bir bilgiyi hatırlamak için, birçok RNN modeli bu bilgiyi hatırlayamaz. LSTM, bu problemi çözmek için özel bir hücre yapısı kullanır. Bu hücre yapısı, giriş, çıkış ve unutma kapıları (input, output, forget gates) adı verilen mekanizmaları içerir. Bu kapılar, hücredeki bilgi akışını kontrol eder ve hangi bilgilerin hafızada saklanacağını ve hangilerinin unutulacağını belirler. LSTM'nin ana bileşenleri şunlardır: Hücre Durumu (Cell State), Giriş Kapısı (Input Gate), Unutma Kapısı (Forget Gate) ve Çıkış Kapısı (Output Gate). Hücre Durumu, hücrenin içinde bulunan uzun vadeli belleği temsil eder. Giriş Kapısı, hangi bilgilerin güncelleneceğini belirlerken Unutma Kapısı, hangi bilgilerin unutulacağını belirler. Çıkış Kapısı ise hücreden çıkan bilgiyi kontrol eder. Bu mekanizmalar sayesinde, LSTM modeli uzun vadeli bağımlılıkları hatırlayabilir ve sıralı verilerde daha etkili bir şekilde çalışabilir. Özellikle dil işleme ve zaman serisi analizi gibi alanlarda başarıyla kullanılır. Bu projede LSTM kullanılmasının ise asıl sebepleri Uzun vadeli bağımlılıkları modelleyebilme yeteneği, çok yaygın olarak kullanılması ve kullanımının kolay olmasıdır.

* + 1. **Modelin eğitimi:**

Başlangıçta modelin eğitimi için gereken tüm bileşenler hazırlanır. İlk adım ise veri setinin yüklenmesi ve hazırlanmasıdır. Bu adımda görüntülerin ve ilgili metin açıklamalarının işlenir, uygun bir yapıya dönüştürülür ve daha sonra modelin eğitiminde kullanılmak üzere uygun formatta saklanırlar. Örneğin, görüntülerin boyutlarının yeniden boyutlandırılması gibi işlemler de bu adımda yapılır. Daha sonra, kararlaştırılmış eğitim parametreleri yüklenir. Bu parametreler arasında, epoch sayısı, öğrenme oranı, batch boyutu ve kayıp fonksiyonu gibi değerler bulunur. Bu parametreler, eğitim sürecinin nasıl ilerleyeceğini ve modelin nasıl optimize edileceğini belirler. Eğitim süreci, veri seti üzerinde yapılan iterasyonlarla başlar. Her bir iterasyonda, model, batch sayısı boyutundaki bir veri grubunu alır ve ileriye doğru geçiş yapar. İleriye doğru geçiş sırasında, modelin tahminlerini ve gerçek etiketler arasındaki kaybı hesaplamak için belirlenen kayıp fonksiyonu kullanılır. Ardından, geriye doğru geçiş yaparak gradyanları hesaplar ve modelin ağırlıklarını günceller. Her epoch sonunda, modelin performansı değerlendirilir. Bu, eğitim ve doğrulama veri setleri üzerinde yapılır. Eğitim kaybı ve doğrulama kaybı gibi değerler kullanılarak modelin ne kadar iyi öğrendiği ve genelleme yeteneğinin ne kadar olduğu değerlendirilir. Eğitim sürecinin sonunda, eğitilmiş modelin durumu kaydedilir ve sonuçlar raporlanır. Bu raporlar, eğitim ve doğrulama kayıpları, doğruluk oranları ve diğer ilgili verileri içerir (Elde edilen sonuçlar, literatürde daha önce yapılan benzer çalışmaların sonuçlarıyla karşılaştırılarak yorumlanabilir, bu karşılaştırma, çalışmanın yenilikçiliği ve etkinliği hakkında daha fazla içgörü sağlayacaktır)(Örnek olarak 2 epoch kullanıldığı durumdaki kayıp değeri 3.2000 civarı bir değer ike epoch 10 olduğunda bu değer 2.0000 civarındadır). Bu süreç, modelin başlangıçtan sona kadar tutarlı bir şekilde eğitilmesini, değerlendirilmesini ve sonuçlarının yorumlanmasını sağlar.

* + 1. **Eğitilen modelin kullanımı:**

İlk olarak, kullanıcı tarafından seçilen görsel dosyası alınır ve işlenebilir bir formata dönüştürülür. Bu dönüşüm işlemi, görselin boyutunu ve formatını uygun hale getirir, böylece model tarafından işlenebilir. Ardından, trainer.py dosyası ile eğitilmiş olan model dosyası yüklenir. Bu model, daha önce Flickr8k veri seti üzerinde eğitilmiş ve görsel özellikler ile açıklamalar arasındaki bağlantı öğretilmiştir. Model, görüntüyü temsil eden özellikleri çıkarmak amacıyla bir CNN modeli olan Inception V3 modelini kullanır. Bu özellikler, görselin içeriğini temsil eder ve ardından bir RNN modeli olan LSTM modeli kullanılarak açıklama oluşturmak için gerekli olan kaynak olarak hizmet eder. Eğitilmiş model, görselin içeriğini anlamak ve uygun bir açıklama üretmek için bu görselden özellikler çıkarır. Bu özellikler daha sonra işlenir ve anlamlı bir cümle oluşturulmaya çalışılır(Şuan ki modelin yeterli eğitime sahip olmaması yüzünden çalışılır başarılıda olunabilir başarısızda). Model, görselin içeriğini tanımlamak ve uygun bir dil yapısı kullanarak anlamlandırmak için önceki eğitiminden elde ettiği bilgileri kullanır. Model, LSTM modelini beslemek için görsel özelliklerle başlar ve ardından her bir adımda LSTM modelinin çıktısını alarak cümle oluşturmaya devam eder. Her adımda, model bir sonraki kelimeyi tahmin eder ve tahmin edilen kelimeyi giriş olarak kullanarak bir sonraki adıma geçer. Bu süreç, gerekli sonlandırma koşulları sağlanana kadar devam eder. Bu şekilde, önceden eğitilmiş modelin kullanılmasıyla, kullanıcının seçtiği görsellerden o görsellerde ne olduğunu anlatan açıklamalar üretilir.

* 1. **Bölüm 3 – Bulgular ve Tartışma**

Uygulamaya yapılan testler sonucunda nazik değerlendirme ile %85lik, ciddi bir değerlendirme ile ise %20 lik bir doğruluk oranı bulunmaktadır.

çizgi film, metin, çocukların yaptığı resimler içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, kırpıntı çizim, çizgi film içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Şekil 1.4.** Uygulamanın ana ekranı ve cevap ekranı

* + 1. **Durum**

Ancak oluşturulan cümlelerin yapısı büyük oranda sorun içermektedir. Sorunun ise modeli eğitirken epoch değerinin ufak tutulması olduğu düşünülmektedir. Böyle düşünülmesinin bazı temel nedenleri var ilki yapılan araştırmalar sonucunda Flickr8k boyutundaki bir veriseti için en uygun epoch değerinin 20-50 arasında olması gerekteğini belirten kaynaklar ile karşılaşıldı ama şuan ki modelin epoch değeri 10. İkinci olarak ise bundan önce kullanılan modelin epoch değeri 2 idi ve o modeli kullanırken durum çok daha kötüydü yani epoch değeri yükseldikçe bu konuda daha düzgün çıktılar alınacaktır. Ancak epoch değerinin aşırı arttırılmasının da bazı sorunları vardır bunlardan en büyük olanı overfitting dir. Daha açıklayıcı olmak gerekirse modelin bir obje ile ilgili gereğiden fazla ayrıntıya sahip olduğu için aynı türde obje olsa bile ufak farkları yüzünden objeyi tanıyamaması durumudur ancak şuan ki durumda overfitting durumuna yakın bile değiliz. Diğer bir sorunda hafıza alanının tüketilmesi bu cidden bir sorun olarak kabul edilebilir çünkü batch değerini 32 den 64 çıkarınca pc her seferinde 1-2 saat çalışıp kapandı yani hafıza problemine sebep olma riski her ne kadar ufakta olsa var.

* + 1. **Ne yapılabilir**

Uygulamayı geliştirmek için öncelikle epoch değeri arttırılarak ve testlere devam edilerek optimum değer bulunulup ona göre değiştirilebilir, bunun dışında bu projede sözcük tabanlı bir derin öğrenme modeli eğitilmiştir ama yapılan araştımalar [10] sonucunda altsözcük tabanlı olarak eğitilen derin öğrenme modellerinin daha doğru sonuçları daha düzgün cümleler halinde verdiği görülmüştür. Bu nedenle eğitilen derin öğrenme modelini altsözcük tabanlı olacak şekilde yeniden eğitilebilir ve diğer kısımlar da buna uygun olarak güncellenebilir. Bunlar hariç uygulamanın bazı durumlarda dropout yapması sağlanabilinir gibi bazı direk olarak şuan ki uygulamanın performansını arttıracak yeniliklerin yanısıra TasvirEt gibi türkçe verisetleri de kullanılarak ikinici bir model daha eğitilip uygulamaya Türkçe dili de eklenebilir. Bunun dışında kullanıcılara verisetine yeni veri ekleme imkanı verilebilir, örneğin kullanıcı istediği görseli yükleyip bu görselde ne olduğunu tanımladıktan sonra veriseti güncellenecektir hem bu özellik trainer.py dosyasının kullanıcının da daha çok işine yaramasını sağlayacaktır çünkü şuan ki versiyonu sadece tokenizer yada modele zarar gelmesi durumunda aynı modeli yeniden üretmek dışında bir amaçla kullanılamaz ama bu özelliğin eklenmesinden sonra tüm kullanıcılar(Bunu yapacak kişinin zaten okuma yazma bilgisinin olması gerektiği için genel olarak ebeveynlerin vb. bunu yapabilecektir) tamamen kendilerine özgü bir modele sahip olabileceklerdir. Bunlar dışında bir ses oluşturma modeli de bulunabilirse yada oluşturulursa proje için yeni yollar açılacaktır.

* 1. **Bölüm 4 – Sonuçlar**

Sonuç olarak proje ciddi değerlendirme yeterince yüksek bir başarı elde edene kadar küçük çocukların kullanımına sunulmayacağı için şimdilik kişisel(Aslında asıl amacını yerine getirdi asıl amacı bir verisetini işlemek, bir derin öğrenme modelini eğitmek, bu modelin image captioning işlemini yerine getirebilmesi ve sonuçların değerlendirilmesiydi ve proje bunların hepsini yerine getirdi ama ben bu projenin sadece yapacağını söylediği şeyleri yapması değil ayrıca projenin gerçek hayatta bir işe yaramasını istiyordum bu nedenle benim açımdan sadece yarı başarılı ancak projenin değerlendirilmesinde benim kişisel amacımın bir katkısı veya zararı bulunamaz) amacını yerine getiremez. Ancak proje başvuru raporunda yapacağını söylediği şeyleri yapmış ve eğitimde öğrencinin bu tür kendi kendisinin de dahil olacağı interakif bir eğitim modeli konusu ile ilgili soru sorulan herkesin desteğini almış ve nasıl çalışacağını anlamayan kişilerin bile merağını cezbetmiştir. Ve uygulama doğruluk oranları ile sorun yaşasa bile ana sebep modelin eğitimi sırasında yaşanan sistemsel (Benim cihazımın sistemi genel değil) ve zamansal sorunlardır. Uygulamaya modelini eğitmesi için gerekli zaman verildiği sürece çok daha iyi bir oran ortaya çıkaracaktır. Bu sebeplerden dolayı hem dediklerini tamamlamış hem de daha fazlasını başarmaya çalışan bu proje bir başarıdır.

1. **KAYNAKLAR**

Kaynakları metin içerisinde göstermede, numara ile kaynak gösterimi yaklaşımı ve IEEE Referans Stili kullanılır. Numara ile kaynak gösteriminde kaynaklar, metin içinde geçtikleri sıraya göre köşeli parantez [ ] içinde numaralandırılır. Daha önce numara verilen kaynağa tekrar atıfta bulunmak istenirse önceki numarası kullanılır. Kaynaklar, kaynaklar kısmında 1. kaynaktan başlayarak numara sırasına göre sıralanır. **Kaynaklar bölümünde yer alan çalışmaların tamamı metin içinde atıflanmalıdır.**

Kaynaklar; metin içerisinde aşağıdaki şekillerde numaralandırılır:

* [1] , 1 nolu kaynak,
* [1-4 ], 1 ve 4 arası (1, 2, 3 ve 4 nolu) kaynaklar,
* [1, 4 ] , 1 ve 4 nolu kaynaklar,
* [1,3-7], 1, 3, 4, 5, 6 ve 7 nolu kaynaklar.

Kaynakların yazımı aşağıdaki kalıplara uygun olmalıdır:

1. Yazarların tamamı kaynakta yer almalıdır.
2. **Yazarların soyadlarının tamamı**, **adlarının ilk harfleri** yazılmalıdır.
3. Dergi ve Konferans isimlerinde kısaltma yapılmamalıdır.
4. Kitap, Dergi ve Bildiri Kitaplarının (Proceedings) isimleri italik olarak yazılmalıdır.
5. Kaynak internet ortamında ise, yukarıdaki verilen bilgilere ek olarak bulunduğu
6. URL adresi (erişim tarihi) verilmelidir.

Yayın bilgilerinin yazımında aşağıdaki sıralama kuralı takip edilir:

Dergide yayın için:

Yazar adının baş harfi. Yazar soyadı, “Tırnak içinde Makale adı”, *Derginin adı*, varsa cilt no(varsa parantez içinde sıra no), sayfa numaraları, yıl.

Konferans için:

Yazar adının baş harfi. Yazar soyadı, “Tırnak içinde Bildiri adı”, *Konferans adı*, varsa cilt no(varsa parantez içinde sıra no), varsa sayfa numaraları, yıl.

Kitap için:

Yazar adının baş harfi. Yazar soyadı, *Kitap adı*, varsa cilt no(varsa parantez içinde sıra no), Yayıncı, yıl.

* 1. **Kullanılan Kaynaklar**

[1] URL: <https://www.businessofapps.com/data/education-app-market/> (Erişim tarihi: 02.06.2024)

[2] URL: https://github.com/HeliosX7/image-captioning-app(Erişim tarihi: 02.06.2024)

[3] URL: https://github.com/purveshpatel511/imageCaptioning(Erişim tarihi: 02.06.2024)

[4] URL: https://github.com/angeligareta/image-captioning(Erişim tarihi: 02.06.2024)

[5] URL: <https://github.com/nithintata/image-caption-generator-using-deep-learning> (Erişim tarihi: 02.06.2024)

[6] URL: https://github.com/Sajid030/image-caption-generator(Erişim tarihi: 02.06.2024)

[7] URL: <https://www.abcmouse.com/abc/?8a08850bc2=T534721651.1717374099.9229> (Erişim tarihi: 02.06.2024)

[8] URL: https://learn.khanacademy.org/khan-academy-kids/(Erişim tarihi: 02.06.2024)

[9] URL: https://lingokids.com/ (Erişim tarihi: 02.06.2024)

[10]URL:<https://basakbuluz.medium.com/makale-okumalar%C4%B1-volm-3-a8c8bc94456f> (Erişim tarihi: 02.06.2024)

URL: https://extremecouponingmom.ca/13-best-free-educational-entertaining-apps-toddlers/(Erişim tarihi: 02.06.2024)

URL: https://github.com/anunay999/image\_captioning\_vgg16(Erişim tarihi: 02.06.2024)

URL: https://ayyucekizrak.medium.com/deri%CC%87n-bi%CC%87r-kar%C5%9Fila%C5%9Ftirma-inception-res-net-versiyonlar%C4%B1-f5cfb83df131(Erişim tarihi: 02.06.2024)

URL: https://bilisim.io/2018/01/22/ogrenim-transferi-keras-ile-inception-v3-kullanimi/(Erişim tarihi: 02.06.2024)

URL: https://paperswithcode.com/method/inception-v3(Erişim tarihi: 02.06.2024)

URL: https://mfakca.medium.com/lstm-nedir-nas%C4%B1l-%C3%A7al%C4%B1%C5%9F%C4%B1r-326866fd8869(Erişim tarihi: 02.06.2024)

URL: https://chatgpt.com/(Erişim tarihi: 02.06.2024)

URL:https://www.youtube.com/watch?v=y2BaTt1fxJU&list=PLCJHEFznK8ZybO3cpfWf4gKbyS5VZgppW(Erişim tarihi: 02.06.2024)

URL: https://www.youtube.com/watch?v=PpEApjWmP-E(Erişim tarihi: 02.06.2024)

URL: https:https://www.youtube.com/watch?v=y2BaTt1fxJU(Erişim tarihi: 02.06.2024)

URL: https://www.youtube.com/watch?v=fUSTbGrL1tc(Erişim tarihi: 02.06.2024)

URL: https://www.youtube.com/watch?v=aaP7JJZuvGs(Erişim tarihi: 02.06.2024)

URL: https://www.youtube.com/watch?v=3e4hsHLDHZA(Erişim tarihi: 02.06.2024)

URL: https://github.com/angeligareta/image-captioning(Erişim tarihi: 02.06.2024)

URL: https://github.com/nithintata/image-caption-generator-using-deep-learning(Erişim tarihi: 02.06.2024)

* 1. **Ekler**

**o170420025\_YigitAliKURT\_GraduationProject.py**

import os

import pickle

import numpy as np

import tkinter as tk

from tkinter import filedialog, messagebox

from PIL import Image, ImageTk

import tensorflow as tf

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

from tensorflow.keras.applications.inception\_v3 import InceptionV3, preprocess\_input

from tensorflow.keras.layers import GlobalAveragePooling2D

from tensorflow.keras.models import Model

class ImageCaptioningApp:

def \_\_init\_\_(self, root):

self.root = root

self.root.title("Second School")

self.root.geometry("400x600")

# Set window icon

self.root.iconphoto(True, tk.PhotoImage(file="Resources/logo.png"))

self.create\_widgets()

self.inception\_model = self.load\_inception\_model()

self.model = load\_model('Resources/image\_captioning\_model.keras')

self.tokenizer = self.load\_tokenizer('Resources/tokenizer.pkl')

def create\_widgets(self):

self.canvas = tk.Canvas(self.root, width=400, height=600)

self.canvas.pack(fill="both", expand=True)

self.bg\_image = Image.open("Resources/main\_bg.jpg")

self.bg\_image = self.bg\_image.resize((400, 600), Image.LANCZOS)

self.bg\_photo = ImageTk.PhotoImage(self.bg\_image)

self.canvas.create\_image(0, 0, image=self.bg\_photo, anchor="nw")

self.photo\_button\_img = Image.open("Resources/photo\_bg.jpg")

self.photo\_button\_img = self.photo\_button\_img.resize((150, 150), Image.LANCZOS)

self.photo\_button\_photo = ImageTk.PhotoImage(self.photo\_button\_img)

self.caption\_button\_img = Image.open("Resources/pencil\_bg.jpg")

self.caption\_button\_img = self.caption\_button\_img.resize((150, 150), Image.LANCZOS)

self.caption\_button\_photo = ImageTk.PhotoImage(self.caption\_button\_img)

self.okay\_button\_img = Image.open("Resources/okay\_bg.png")

self.okay\_button\_img = self.okay\_button\_img.resize((160, 160), Image.LANCZOS)

self.okay\_button\_photo = ImageTk.PhotoImage(self.okay\_button\_img)

self.image\_button = tk.Button(self.root, image=self.photo\_button\_photo, command=self.browse\_image, bd=0)

self.image\_button\_window = self.canvas.create\_window(200, 200, window=self.image\_button)

self.generate\_button = tk.Button(self.root, image=self.caption\_button\_photo, command=self.generate\_caption, bd=0)

self.generate\_button\_window = self.canvas.create\_window(200, 380, window=self.generate\_button)

def load\_tokenizer(self, tokenizer\_path):

with open(tokenizer\_path, 'rb') as f:

tokenizer = pickle.load(f)

return tokenizer

def load\_inception\_model(self):

inception\_model = InceptionV3(include\_top=False, weights='imagenet')

new\_input = inception\_model.input

hidden\_layer = inception\_model.layers[-1].output

pooling\_layer = GlobalAveragePooling2D()(hidden\_layer)

return Model(inputs=new\_input, outputs=pooling\_layer)

def browse\_image(self):

image\_file = filedialog.askopenfilename(filetypes=[("Image Files", "\*.jpg;\*.jpeg;\*.png")])

self.image\_path = image\_file

def preprocess\_image(self, image\_path):

img = load\_img(image\_path, target\_size=(299, 299))

img = img\_to\_array(img)

img = np.expand\_dims(img, axis=0)

img = preprocess\_input(img)

return img

def extract\_image\_features(self, image\_path, model):

img = self.preprocess\_image(image\_path)

feature = model.predict(img, verbose=0)

return feature

def generate\_sentence\_from\_features(self, model, tokenizer, img\_feature, max\_length):

in\_text = ''

words = []

for \_ in range(max\_length):

sequence = tokenizer.texts\_to\_sequences([in\_text])[0]

sequence = pad\_sequences([sequence], maxlen=max\_length)

yhat = model.predict([img\_feature, sequence], verbose=0)

yhat = np.argmax(yhat)

word = tokenizer.index\_word.get(yhat)

if word is None:

break

if len(in\_text) == 0:

word = word.capitalize()

in\_text += word + ' '

if word == '<end>':

break

words.append(word)

if len(' '.join(words)) > 29:

in\_text += '\n'

words = []

return in\_text.strip()

def generate\_caption(self):

if not hasattr(self, 'image\_path') or not self.image\_path:

self.show\_caption\_message("No Image Selected", "")

return

feature = self.extract\_image\_features(self.image\_path, self.inception\_model)

caption = self.generate\_sentence\_from\_features(self.model, self.tokenizer, feature, 34)

self.show\_caption\_message(os.path.basename(self.image\_path), caption)

def show\_caption\_message(self, image\_name, caption):

self.caption\_window = tk.Toplevel(self.root)

self.caption\_window.title("Second Teacher")

self.caption\_window.geometry("650x450")

bg\_image = Image.open("Resources/output\_bg.png")

bg\_image = bg\_image.resize((650, 450), Image.LANCZOS)

bg\_photo = ImageTk.PhotoImage(bg\_image)

canvas = tk.Canvas(self.caption\_window, width=650, height=450)

canvas.pack(fill="both", expand=True)

canvas.create\_image(0, 0, image=bg\_photo, anchor="nw")

caption\_text = tk.Text(self.caption\_window, height=5, width=30, font=("Segoe Print", 16))

caption\_text\_window = canvas.create\_window(550, 50, window=caption\_text, anchor="ne")

caption\_text.insert(tk.END, f'{caption}\n')

caption\_window\_okay\_button = tk.Button(self.caption\_window, image=self.okay\_button\_photo, command=self.close\_caption\_window, bd=0)

caption\_window\_okay\_button\_window = canvas.create\_window(550, 350, window=caption\_window\_okay\_button)

self.caption\_window.bg\_photo = bg\_photo

def close\_caption\_window(self):

self.caption\_window.destroy()

if \_\_name\_\_ == "\_\_main\_\_":

root = tk.Tk()

app = ImageCaptioningApp(root)

root.mainloop()

**trainer.py**

import os

import pickle

import numpy as np

import pandas as pd

import tensorflow as tf

from tensorflow.keras.layers import Embedding, LSTM, Dense, Flatten, RepeatVector, concatenate, GlobalAveragePooling2D

from tensorflow.keras.models import Model

from tensorflow.keras.preprocessing.image import load\_img, img\_to\_array

from tensorflow.keras.applications.inception\_v3 import InceptionV3, preprocess\_input

from tensorflow.keras.preprocessing.text import Tokenizer

from tensorflow.keras.preprocessing.sequence import pad\_sequences

from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

embedding\_dim = 256

units = 512

max\_length = 34

vocab\_size = 5000

def load\_inception\_model():

inception\_model = InceptionV3(include\_top=False, weights='imagenet')

new\_input = inception\_model.input

hidden\_layer = inception\_model.layers[-1].output

pooling\_layer = GlobalAveragePooling2D()(hidden\_layer)

return Model(inputs=new\_input, outputs=pooling\_layer)

image\_features\_extract\_model = load\_inception\_model()

def load\_captions(file\_path):

df = pd.read\_csv(file\_path)

captions = df.groupby('image')['caption'].apply(list).to\_dict()

return captions

def preprocess\_image(image\_path):

img = load\_img(image\_path, target\_size=(299, 299))

img = img\_to\_array(img)

img = np.expand\_dims(img, axis=0)

img = preprocess\_input(img)

return img

def extract\_image\_features(directory):

features = {}

print("Image extraction started\n")

for img\_name in os.listdir(directory):

img\_path = os.path.join(directory, img\_name)

img = preprocess\_image(img\_path)

feature = image\_features\_extract\_model.predict(img, verbose=0)

features[img\_name] = feature

print("Image extraction finished\n")

return features

def tokenize\_captions(captions):

tokenizer = Tokenizer(num\_words=vocab\_size, oov\_token="<unk>")

tokenizer.fit\_on\_texts([caption for captions\_list in captions.values() for caption in captions\_list])

return tokenizer

def create\_sequences(tokenizer, max\_length, captions, features):

X1, X2, y = [], [], []

for key, caption\_list in captions.items():

print("Create sequence step: " + key + "\n")

feature = features[key][0]

for caption in caption\_list:

seq = tokenizer.texts\_to\_sequences([caption])[0]

for i in range(1, len(seq)):

in\_seq, out\_seq = seq[:i], seq[i]

in\_seq = pad\_sequences([in\_seq], maxlen=max\_length)[0]

out\_seq = tf.keras.utils.to\_categorical([out\_seq], num\_classes=vocab\_size)[0]

X1.append(feature)

X2.append(in\_seq)

y.append(out\_seq)

return np.array(X1), np.array(X2), np.array(y)

def define\_model(vocab\_size, max\_length):

inputs1 = tf.keras.Input(shape=(2048,))

fe1 = Dense(embedding\_dim, activation='relu')(inputs1)

fe2 = RepeatVector(max\_length)(fe1)

inputs2 = tf.keras.Input(shape=(max\_length,))

se1 = Embedding(vocab\_size, embedding\_dim, mask\_zero=True)(inputs2)

se2 = LSTM(units, return\_sequences=True)(se1)

decoder1 = concatenate([fe2, se2])

decoder2 = LSTM(units)(decoder1)

outputs = Dense(vocab\_size, activation='softmax')(decoder2)

model = Model(inputs=[inputs1, inputs2], outputs=outputs)

model.compile(loss='categorical\_crossentropy', optimizer='adam')

return model

def train\_model(image\_dir, captions\_file, model\_save\_path, tokenizer\_save\_path):

captions = load\_captions(captions\_file)

image\_features = extract\_image\_features(image\_dir)

tokenizer = tokenize\_captions(captions)

X1, X2, y = create\_sequences(tokenizer, max\_length, captions, image\_features)

model = define\_model(vocab\_size, max\_length)

checkpoint = ModelCheckpoint(model\_save\_path, monitor='loss', save\_best\_only=True, mode='min')

early\_stopping = EarlyStopping(monitor='loss', patience=3)

model.fit([X1, X2], y, epochs=12, verbose=2, callbacks=[checkpoint, early\_stopping])

print(f'Saving tokenizer to {tokenizer\_save\_path}')

with open(tokenizer\_save\_path, 'wb') as f:

pickle.dump(tokenizer, f)

train\_model('Flickr8k/Images/', 'Flickr8k/captions.txt', 'Resources/image\_captioning\_model.keras', 'Resources/tokenizer.pkl')

input("Training complete. Press Enter to exit.")

**Özgeçmiş (CV)(**Asıl cv çok güzel bir formatta idi ama buraya yapıştırınca bozuldu**)**

HAKKIMDA

Daha fazlasını yapabileceğine inanan ve kendini geliştirmek isteyen bir üniversite öğrencisiyim.

İLETİŞİM

Yiğit Ali KURT 05553239457

yigitali.yilmaz@hotmail.com

eragonpyrus@gmail.com

ADRES

Yakuplu mah. 30. sok 9/11 Beylikdüzü/İstanbul

GEÇMİŞ KATILIMLARI

2018 Yılında robotik yarışmalarına katıldı Çeyrek finalde elendi.

2019 Yılında robotik yarışmalarına katıldı Yarı finalde elendi(Hakem hatası sebebiyle)

2019 Yılında Ulusal Bilim Olimpiyatlarına Katıldı (İlk aşamayı başarıyla geçti ikinci aşamada elendi)

HOBİLER

Kitap okumak

Teknolojideki yenilikleri araştırmak Tarihi öğrenmek

Yeni bir şey oluşturmak(Sanal bir uygulama, bir maket vb.)

Yemek yapmak ve yemek

Bilgisayar oyunları

Bowling

BECERİLER VE YETERLİLİKLER

Robotik alanında tasarım ve montaj konusunda deneyimli

İngilizce (B2 seviyesi konuşma, C1 seviyesi okuma ve yazma)

Robotik kodlama alanında deneyimli

Web kodlama alanında orta düzey(Html ,CSS ,JS kullanımı)

Python, Java,C dillerinde orta düzey

Ekip yönetme deneyimi var.

EĞİTİM GEÇMİŞİ

Ambarlı İlkokulunda 1. - 4 . Sınıflarda

Kemal Arıkan Ortaokulunda 4 .- 7. Sınıflarda

Koç Ortaokulunda 8. Sınıfta

Hüseyin Yıldız Anadolu Lisesinde 9. - 12. Sınıflarda

Eğitim Aldı

Şuanda ise:

Marmara Üniversitesi Teknoloji Fakültesi

Bilgisayar Mühendisliği 4. Sınıf Öğrencisi

Ayrıntılı Açıklama:

Liseye başladığım yıldan (2016-2017) itibaren yazılıma merak saldım ve bu alanda kendimi geliştirmeye başladım. Öncelikle yapılmakta olan ulusal olimpiyatlara da katılmak amacıyla C ve C++ dillerini öğrenmeye başladım. Bu dönemde ayrıca Adobe Photoshop, GIMP, Paint.net gibi görsel işleme ve Camtasia gibi video işleme programlarında temel düzeyde beceri kazandım.

İkinci senemde (2017-2018) çalışmakta olduğum dillerde kendimi geliştirmeye devam ettim ve temel düzeyde Html/CSS ve Orta düzeyde otonom bir robot kodu yazabilecek düzeyde arduino programlama öğrendim. Ayrıca bu dönemde ilk kez Ulusal bilim olimpiyatlarına ve İstanbul bölgesel bilim olimpiyatlarına bilgisayar dalında katıldım ama başarısız oldum. Bunlar dışında yapılan bir robotik yarışmasına okulumu temsil etmek amacıyla katıldım, yarışma için yapılan minisumo robotunun yazılımının sorumluluğu üstlendim. Yapılan robot çeyrek finallerde kablolarının yanması sonucu elendi.

Üçüncü senemde (2018-2019) çalışmakta olduğum dillerde kendimi geliştirmeye devam ettim. Bunun dışında temel düzeyde roket tasarımı üzerinde çalıştım. Ayrıca Ulusal bilim olimpiyatlarında bilgisayar alanında yeniden katıldım ve ön elemeleri geçip eğitim kampına davet edildim. Bunlara ek olarak bu dönemdeki robotik yarışmasına tekrardan katıldım ve bu sefer yapılan minisumo robotunun yapımının tüm sorumluluğunu üstlendim. Yapılan robot yarı finallerde hakem hatası sebebiyle elendi.

Dördüncü senemde(2019-2020) büyük oranda üniversite sınavında hazırlık içinde geçti. Bunun dışında ulusal bilim olimpiyatlarının eğitim kamplarına katıldım ve alanda devam ettim ama kişisel sebeplerden ötürü ulusal olimpiyatların son aşamasında elendim.

Ayrıca bu dönemde okulun daha alt sınıftaki ama yazılıma merakı olan öğrencilerine mentorluk yaptım.

Üniversiteye başladığım yıldan (2020-2021) itibaren kendimi yazılım alanında geliştirmek amacıyla temel düzeyde python ve java öğrendim. Ayrıca üniversitenin roket kulübüne katıldım ve orada yazılım alanında çalışmaya başladım ama kısa sürede kişisel sebeplerden ötürü ayrılmaya karar verdim. Bunlar dışında kod yazma biçimimi daha bütünleşmiş bir biçimden daha sınıflandırılmış ve parçalara ayrılmış biçime çevirmeye başladım.

İkinci senemde (2021-2022) covid-19 un da yayılması ile daha az etkinlikle ilgilendim ama üniversitede verilen ödevler ve projeler sayesinde daha çok tecrübe kazandım ve daha sistematik çalışmaya başladım ve java ve python alanında kendimi daha da geliştirdim. Bunlar dışında bazı özel istekler sonucu python ile görüntü işleme konusunda da ayrıca çalıştım.

Üçüncü senemde (2022-2023) aynı şekilde covid-19 un son dönemleri olması hemde sürekli evde oluması sebebiyle minimal sayıda etkinlik ile ilgilendim. Bunun dışından python ve java alanlarında kendimi geliştirdim ve bu dillerle yazdığım bazı ufak projelerde çalıştım. Bunlar dışında github la ilgili bir şeyler yapmayı denedim ama sonra bıraktım.

Dördüncü senemde (2023-2024) Size bu cv’yi yolladım.