MAKALE RAPORU

Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models

Yaşar ŞENTÜRK 6017023

EE514 DERİN ÖĞRENME

Prof.Dr.Mehmet Kemal GÜLLÜ

İZMİR BAKIRÇAY ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ A.B.D.

Haziran 2023

İÇİNDEKİLER

ŞEKİLLER DİZİNİ 3
1. MAKALE KÜNYESİ 4
2. ÖZET 4
3. GİRİŞ 5
4. MOTİVASYON 5
5. VISUAL ChatGPT' YE GİRİŞ İÇİN GEREKLİ BİLGİLER 6
6. VISUAL ChatGPT' YE GENEL BİR BAKIŞ 8
7. VISUAL ChatGPT 15
8. DENEYLER 17
9. KOD İNCELEMESİ VE UYGULAMA 17
10. SINIRLAMALAR 24
11. SONUÇLAR VE TARTIŞMA 24
12. KAYNAKLAR 26

ŞEKİLLER DİZİNİ

Şekil 1. Temel GPT Modeli 7
Şekil 2. Visual ChatGPT Mimarisi 8
Şekil 3. BLIP Ön Eğitim Modeli 9
Şekil 4. Stable Diffusion Temel Modeli 10
Şekil 5. U-Net Üretici Mimarisi 11
Şekil 6. Ayırıcı Ağ Modeli 12
Şekil 7. Visual ChatGPT Genel Bakışı 13
Şekil 8. Prompt Manager Genel Bakış 15
Şekil 9. Visual ChatGPT Verilen Talimatların Karşılaştırılması 23
Şekil 10. Canny Kenar Tespiti 23

1. MAKALE KÜNYESİ

Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., & Duan, N. (2023). Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. Microsoft Research Asia. arXiv:2303.04671v1[cs.CV].

Anahtar Kelimeler: ChatGPT; VisualChatGPT; Image Processing; Visual Foundation Models; Language Interface.

2. ÖZET

ChatGPT, OpenAI tarafından geliştirilen bir yapay zekâ modelidir. Bu model, büyük miktarda metin verisiyle eğitilmiştir ve dil anlama, dil üretme ve konuşma becerilerine sahiptir. ChatGPT, doğal dil işleme, metin tabanlı soru-cevap ve benzeri görevlerde başarılı bir şekilde kullanılabilmektedir. Ancak, görsel verileri işleme veya üretme yeteneğine sahip değildir.

Bu nedenle "Visual ChatGPT" sistemi geliştirmiştir. Bu sistem, kullanıcının ChatGPT ile etkileşim halinde olduğu bir Görsel Temel Model kullanmaktadır. Böylece sadece dil kullanarak metin tabanlı değil aynı zamanda görselleri gönderip alarak ChatGPT ile iletişim sağlanabilmektedir. Bu sistem birden fazla yapay zekâ modelinin işbirliği yaparak çok adımlı görsel soruları veya görsel düzenleme talimatlarını gerçekleştirmesine olanak tanımaktadır. Ayrıca geri bildirimler alınabilmektedir.

Görsel model bilgisini ChatGPT' ye eklemek amacıyla bir dizi yol tasarlanmıştır. Bu tasarımlar çoklu giriş/çıkış modelleri ve çoklu geri bildirim gerektiren modelleri içermektedir. Yapılan deneylerde Visual ChatGPT' nin Görsel Temel Model yardımıyla ChatGPT'nin görsel yeteneklerini araştırmak için birçok olasılık sunduğu görülmektedir.

3. GİRİŞ

Büyük dil modellerinin gelişmesiyle büyük ilerlemeler kaydedilmiştir. Bu ilerlemelerden biri de ChatGPT' dir. ChatGPT, InstructGPT temel alınarak oluşturulmuş ve kullanıcıyla sohbet tarzında etkileşimde bulunacak tarzda eğitilimiştir. ChatGPT konuşmanın bağlamını koruyabilmekte, soruları takip edebilmekte ve kendisinin ürettiği yanıtları düzeltebilmektedir.

ChatGPT, metin temelli işlemlerde son derece güçlü yeteneklere sahip olsa da görsel işleme yetenekleri sınırlıdır çünkü ChatGPT dil verileriyle eğitilmiştir. Görsel temel modellerinin (Visual Foundation Models-VFMs) ise karmaşık görüntüleri anlama ve üretme yeteneklerinin yüksek potansiyele sahip olduğu bilinmektedir. ChatGPT' nin görsel işlemede sınırlı kabiliyetlerini geliştirmek amacıyla Visual ChatGPT sistemi geliştirmiştir. Bu sistemde mevcut ChatGPT temel alınarak çeşitli Görsel Temel Modelleri entegre edilmiştir. ChatGPT ile Görsel Temel Modellerin arasında kılavuz yöneticisi kullanılmıştır. Kılavuz yöneticisi ChatGPT' ye Görsel Temel Modelin yeteneklerini belirtmek, giriş-çıkış formatlarını belirtmek, görsel bilgilerinin anlaşılması ve dil formatına dönüştürülmesini sağlamak ve farklı Görsel Temel Modellerin kullanımlarını yönetmek amacıyla kullanılmaktadır. Kılavuz yöneticisi kullanılarak ChatGPT, Görsel Temel Modelleri kullanabilmektedir.

Çalışmada ChatGPT ile Görsel Temel Modellerin birleşimini sağlayan Visual ChatGPT hakkında teorik bilgi ve görsel görevlerde kullanımı incelenmiştir.

4. MOTİVASYON

Bu çalışmanın yapıldığı alanda bazı eksiklikler bulunmaktadır ve bu eksiklikler, Visual ChatGPT çalışmasının amacını belirlemektedir. Öncelikle ChatGPT yalnızca dil verilerini işleyebilmekte ve görsel verileri işleme veya üretebilme yeteneğine sahip değildir. Bu durum çalışmanın temel amacı olarak ortaya çıkmaktadır. Ayrıca çok adımlı görsel soruların veya düzenleme talimatlarının gerçekleştirilmesi, görsel temel modellerin (VFMs) tek bir görevde uzmanlaşmış olmaları nedeniyle zorlu bir hal almaktadır. Bu da ihtiyaç duyulan görsel işleme ve dönüşüm yetenekleri arasında bağlantı eksikliğine işaret etmektedir. Dahası dil ve görsel bilgi arasındaki veri etkileşimi konusunda yeterli bilgi birikimi bulunmamaktadır. Son olarak yeni modellerin eğitilmesi için büyük bir hesaplama kaynağı gerekmektedir. Bu nedenle çalışmada ele alınan sorunların çözümü için etkili yöntemlerin geliştirilmesi gerekmektedir.

İncelenen makalenin ana fikri yeni geliştirilmiş olan dil ve görsel arasındaki etkileşimi sağlayan sistem "Visual ChatGPT" yi tanıtmaktır. ChatGPT' nin görsel rollerini araştırmak ve görsel-dil tabanlı görevlerde etkin kullanılabilen bir yapay zekâ modeli oluşturulması ana amaçtır. Bu

amaçla Visual ChatGPT geliştirilmiştir. Visual ChatGPT ile ChatGPT'nin sadece dil işleme yetenekleri geliştirilmekle kalınmamış aynı zamanda görsel bilgi işleme yetenekleri de görsel temel model (VFMs) kullanılarak eklenmiş ve kullanıcının ChatGPT ile etkileşime geçerek yapay zekâ tabanlı dil arayüzü olarak geliştirilmiş yeni bir sistem "Visual ChatGPT" karşımıza çıkmıştır. Bu şekilde karmaşık görsel-dil tabanlı problemleri çözme becerisine sahip olunmuştur.

Çalışmanın başlangıcında çözülmesi gereken problemler büyük ölçüde çözüme kavuşturularak literatüre önemli katkılarda bulunulmuştur.

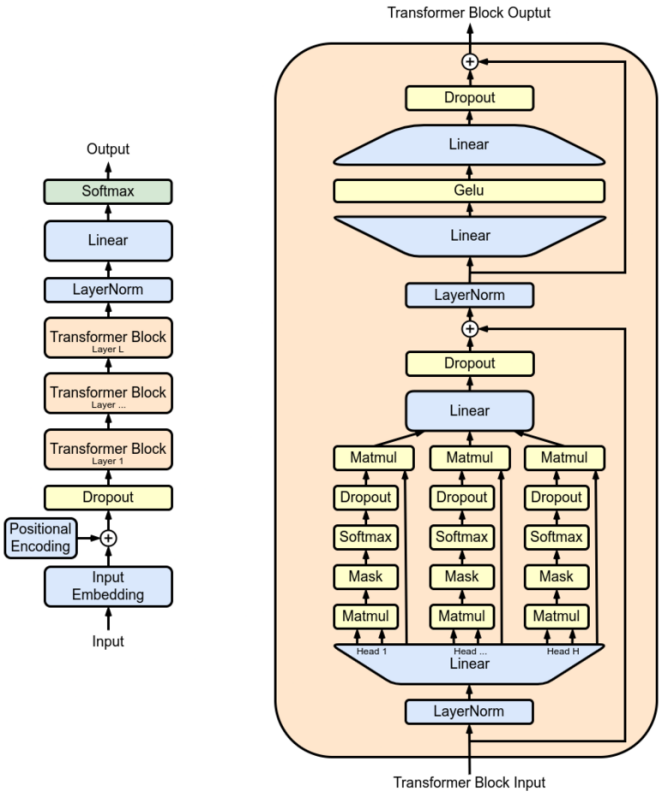
Geliştirilen Visual ChatGPT görsel soru-cevap uygulamaları, sanat tasarım uygulamaları, eğitim araçları, soru-cevap uygulamaları gibi çeşitli alanlarda kendine kullanım alanı bulabilir. Dil ve görsel arasındaki etkileşimi geliştirerek birçok alanda yenilikçi çözümler sunma potansiyeline sahiptir.

5. VISUAL ChatGPT' YE GİRİŞ İÇİN GEREKLİ BİLGİLER

Visual ChatGPT' den bahsedilmeden önce bazı bilgiler hakkında fikir sahibi olmak gereklidir. Visual ChatGPT' nin temeli ChatGPT' ye dayanmasından ötürü bu temel bilgiler ChatGPT' yi kapsamaktadır. ChatGPT temel olarak büyük dil modeli, doğal dil işleme gibi alanlara dayanmaktadır. Bu alanlardaki bilgiler aşağıda verilmiştir.

Büyük dil modeli, derin öğrenme teknikleriyle oluşturulan ve doğal dil işleme alanlarında kullan bir yapay zekâ modelidir. Büyük dil modelleri, metinleri anlama, metin üretimi, metin tabanlı soru-cevap, dil tabanlı uygulamalarda kullanılabilmektedir. Bu modeller büyük miktarda metin verisiyle eğitilmektedir ve eğitim işleminin ardından kullanıcı talimatlarını işleyebilmektedir. Eğitim sürecinde metin yapı ve bağlamlarını öğrenir ve metinleri anlamak-üretmek için gerekli bilgileri elde etmektedir. En popüler büyük dil modelleri olarak günümüzde GPT-3, GPT-4 gösterilmektedir.

Generative Pre-trained Transformer (GPT), doğal dil işleme (NLP) alanında kullanılan bir yapay zekâ modelidir. GPT, OpenAI tarafından geliştirilmiştir ve büyük dil modeliyle önceden eğitilmektedir. Bu modeller transformer mimarisine dayanmaktadır ve metin tabanlı insan benzeri yeni içerik üretme yeteneğine sahiptir. GPT dil anlama, metin oluşturma, çeviri ve diğer doğal dil işleme görevlerinde kullanılabilmektedir. Aşağıdaki şekilde temel GPT modeli görülmektedir.



Şekil 1. Temel GPT Modeli

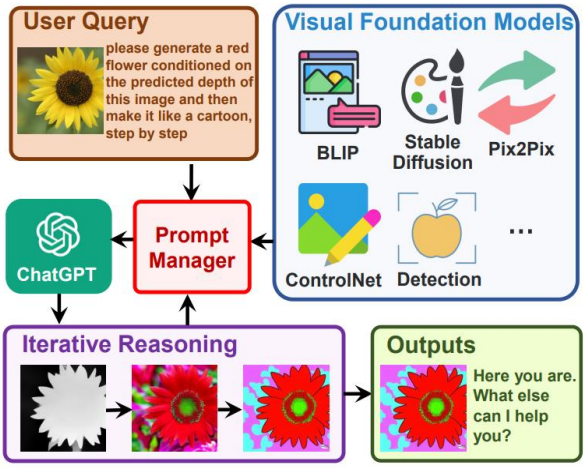
Şekil 1 üzerinden temel GPT modelini şu şekilde açıklayabiliriz. GPT modeli, WEB sayfalarında bulunan metin tabanlı veriler ile eğitilmektedir. Bu WEB sayfalarında bulunan metin tabanlı veriler "input" olarak modele girmektedir. Ardından modele giren veriler analiz edilir ve verideki her bir kelimenin bağlamsal anlamı anlaşılmaya çalışılır. Dropout basamağında gerçekleşen anlamlandırma ve eğitimin ardından Transformer bloklarında giriş verileri işlenerek çıktı üretilir. Her bir transformer bloğu, kendinden önceki bloğun çıktılarını alarak bu çıktıları işleyip yeni çıktı üretmektedir. Transformer bloklarında bu sayede dikkat mekanizması ve self-attention takip edilerek kelimeler bağlamında kullanılarak çıktı üretilir. LayerNorm bloğunda ise Transformer bloklarının çıktıları standartlaştırır. Standartlaştırma işlemi her katmandaki aktivasyonların normalleştirilmesiyle sağlanır. Linear bloğunda, katmanlar arasında boyut değişiklikleri gerçekleştirilir ve matris çarpımları, ağırlıklar ve aktivasyonlar hesaplanarak çıktılar şekillendirilir. Softmax bloğunda ise çıktılar normalleştirilir ve olasılık dağılımı elde edilir. Gelu' dan bahsetmek gerekirse, (Gaussian Error Linear Unit) fonksiyonu modele non-lineerlik eklemek ve performansı arttırmak amacıyla kullanılmaktadır [1].

ChatGPT, OpenAI tarafından geliştirilen doğal dil işleme amacıyla metin tabanlı sohbet olarak kullanılabilen bir yapay zekâ dil modelidir. ChatGPT, bir GPT türevidir. ChatGPT ile GPT' nin tüm özellikleri kullanılarak metin tabanlı girdiler sağlanabilir ve yanıt alınabilir.

6. VISUAL ChatGPT' YE GENEL BİR BAKIŞ

ChatGPT' nin metin-görsel tabanlı işlemlerde eksikliğinin görülmesi üzerine yazarlar, sıfırdan bir ChatGPT oluşturmak yerine çeşitli görsel temel modelleri (VFMs) doğrudan ChatGPT' ye entegre etmeyi önermektedir. Bu VFMs' lerin entegrasyonunu kolaylaştırmak amacıyla yazarlar Prompt Yöneticisi adını verdikleri bir araç tanıtmaktadırlar. Prompt Yöneticisi, her bir VFMs' lerin giriş-çıkış formatlarını belirleme, görsel bilgiyi dil formatına dönüştürme ve farklı VFMs'lerin geçmişlerini, önceliklerini ve çatışmalarını yönetmektedir. Prompt Yöneticisi yardımıyla ChatGPT bu VFMs' leri tekrarlayarak kullanabilir ve kullanıcı gereksinimlerini karşılayana kadar geri bildirİm alabilir.

Görsel temel model (Visual Foundation Model - VFM), görsel bilgileri işlemek ve anlamak amacıyla kullanılan derin öğrenme modelleridir. Görüntü analizi, görüntü sınıflandırma, nesne tanıma gibi görsel görevlerde kullanılmaktadır. Görsel temel modeller karmaşık görüntü verilerini işleyerek özellik çıkarmayı, görüntüleri oluşturmayı ve değiştirmeyi sağlamaktadır. Bu modeller büyük bir veri kümesi üzerinde bazı görevlere için eğitilmektedir. Visual ChatGPT' de görsel temel modeller kullanılarak görüntü işleme ve dönüşümü gerçekleştirilerek ChatGPT' nin görsel bilgilere dayalı görevleri yerine getirmesi sağlanmaktadır.

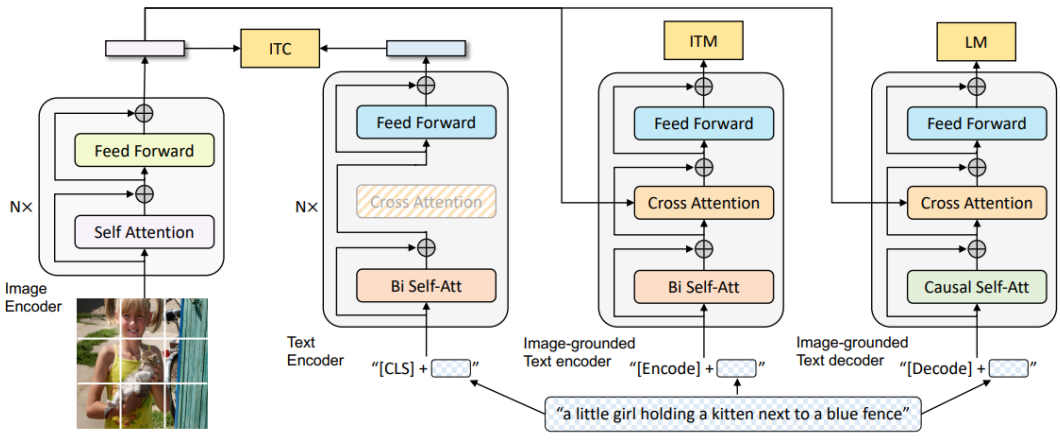


Şekil 2. Visual ChatGPT Mimarisi

Yukarıdaki şekilde Visual ChatGPT' nin mimarisi görülmektedir. Şekilden yola çıkarak Visual ChatGPT işleyişinden bahsedilebilir. Görüldüğü üzere kullanıcı görsel bilgi olarak sarı çiçek resmi girmektedir ve komut olarak "görüntünün tahmini derinliğine bağlı olarak kırmızı bir

çiçek oluştur ve karikatür tarzına çevir" gibi bir karmaşık talimat vermektedir. Prompt Yöneticisi' nin yardımıyla, Visual ChatGPT ile Görsel Temel Modellerin (VFMs) bir yürütme süreci başlatılmaktadır. Bu durumda önce derinlik tahmin modelini uygulayarak derinlik bilgisini tespit edilmektedir. Ardından derinlikten görüntü modelini kullanarak derinlik bilgisine sahip kırmızı bir çiçek figürü oluşturmaktadır ve son olarak Sabit difüzyon modeline dayalı stil transferi VFMs kullanarak bu görüntünün stilini bir karikatüre dönüştürmektedir. Bu işleyiş hattı boyunca Prompt Yöneticisi ChatGPT' ye görsel format türünü sağlayarak ve bilgi dönüşümünün sürecini kaydederek ChatGPT için bir görev yöneticisi olarak hizmet etmektedir [2]. Son olarak, Visual ChatGPT, Prompt Yöneticisi' nden "karikatür" ipuçlarını aldığında işleyiş hattını sonlandırmaktadır ve sonucu göstermektedir. Aşağıda Şekil 2' de Visual ChatGPT mimarisinde Visual Foundation Models kısmında kullanılabilecek modeller incelenmiş ve açıklanmıştır.

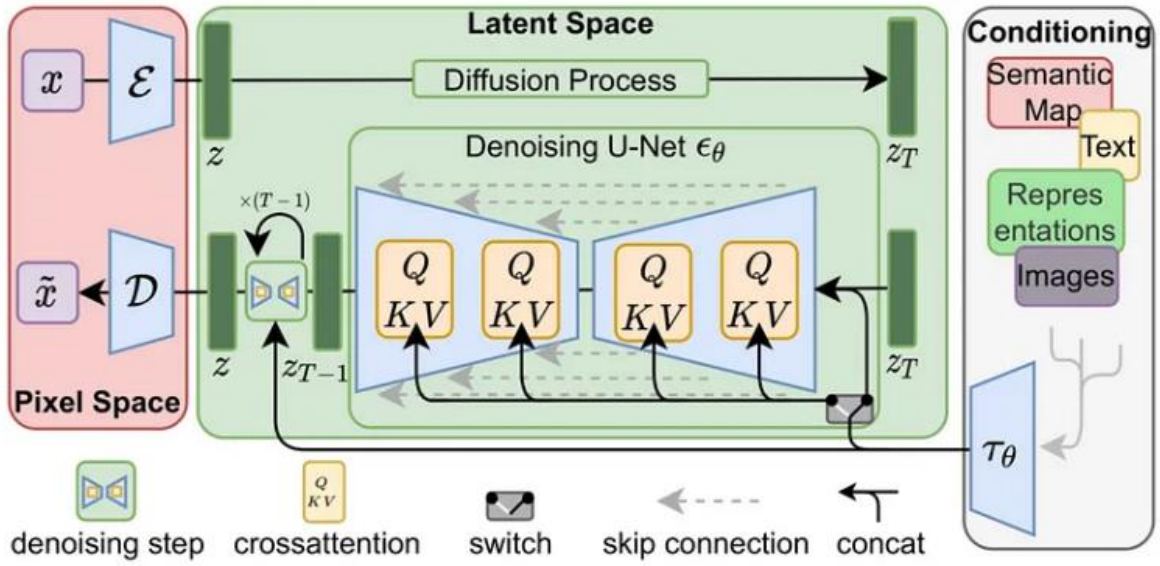
BLIP (Bootstrapping Language-Image Pre-training), dil ve görüntü öncesi eğitimin birleştirildiği bir makine öğrenimi modelidir. BLIP, dil ve görüntü işleme alanlarında başarı elde etmek amacıyla bilgileri bir araya getirmektedir. Genellikle çoklu görev öğrenme yaklaşımını kullanan BLIP, dil ve görsel anlama becerilerini geliştirmektedir. BLIP modelinde büyük ölçekli veri kümeleri üzerinde ön eğitim yapılmaktadır. Dil modelleri için internet üzerindeki metin tabanlı veriler, görüntü kümesi için ise görsel veri tabanları kullanılmaktadır. Ön eğitim tamamlandıktan sonra, BLIP dil ve görüntü işleme görevlerini hedefleyen belirli amaca yönelik(görsel soru cevaplama, görüntü-metin eşleştirme vb.) alanlarda kullanılabilmektedir. Aşağıdaki şekilde ön eğitim model mimarisi ve BLIP hedefleri görülmektedir [3].



Şekil 3. BLIP Ön Eğitim Modeli

Şekil 3’ te bahsedilen model üç farklı işlevde çalışabilmektedir. (1) Tek modalite kodlayıcısı, görüntü ve dil temsillerini hizalamak amacıyla bir görüntü-metin karşıtlı (ITC) kaybıyla eğitilir. (2) Görüntüye dayalı metin kodlayıcısı, görüntü-dil etkileşimlerini modellemek amacıyla fazladan cross-attention katmanları içermektedir ve pozitif-negatif görüntü-metin çiftlerini ayırt etmek için bir görüntü-metin eşleştirme (ITM) kaybıyla eğitilir. (2) Görüntüye dayalı metin dekoderi, çift yönlü self-attention katmanlarını tek yönlü self-attention katmanlarıyla değiştirmektedir ve aynı cross-attention katmanları ve ileri besleme ağlarıyla kodlayıcıyı paylaşmaktadır. Dekoder, görüntülere dayalı açıklamaları üretmek için bir dil modelleme kaybıyla eğitilmektedir.

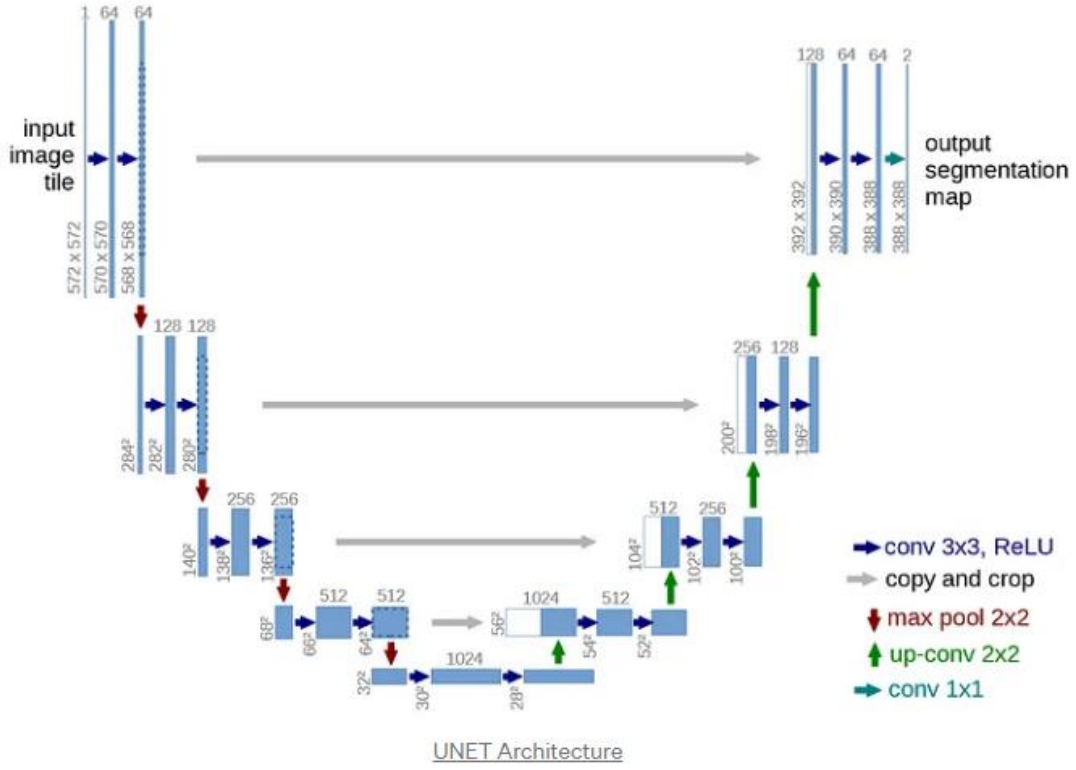
Stable Diffusion, herhangi bir metin girdisi verildiğinde foto gerçekçi görüntüler üretebilen bir metinden görüntüye yayılım modelidir. Bu model ile mevcut tüm model kontrol noktalarına genel bir bakış sunulmaktadır. Gürültü gideren autoencoder ardışık uygulamasıyla bir görüntünün oluşturma sürecini parçalayarak gizli yayılım modellerinin görüntü verilerinde sentez sonuçları elde ettiği modeller olarak açıklanmaktadır. Modellerin genelde doğrudan piksel uzayında çalışması nedeniyle yüzlerce GPU gerekmektedir ancak Stable diffusion modelinde önceden eğitilmiş autoencoders’ ların gizli uzaya yerleştirerek kalite ve esneklikten ödün vermeden sınırlı kaynakla eğitilebilmelerine olanak sağlanmıştır. Günümüzde derin öğrenme alanında önemli bir model olan Stable Diffusion modelinin temel mimarisi aşağıdaki şekilde verilmiştir.



Şekil 4. Stable Diffusion Temel Modeli

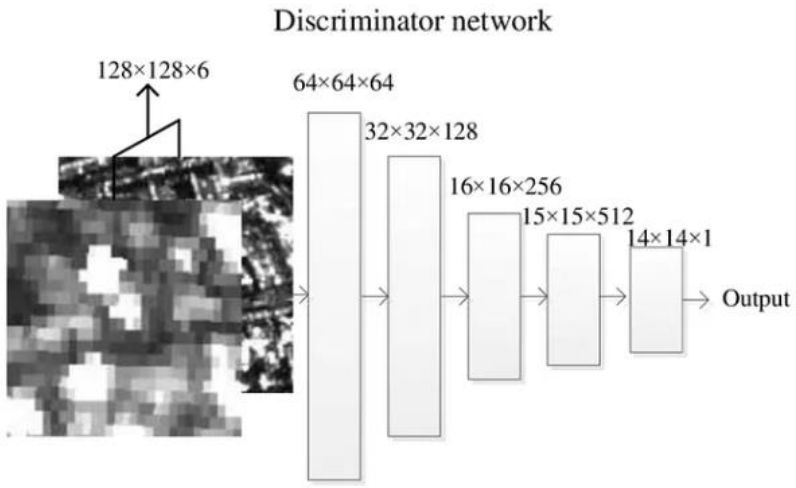
Cross-attention katmanlarını model mimarisine ekleyerek güçlü ve esnek üreteç elde edilmiştir. Bu sayede yüksek çözünürlüklü sentezleme sağlanabilmektedir [4].

Pix2Pix, görüntüden görüntüye çeviri olarak tanımlanabilmektedir. Belirli bir görev amacıyla özel modellere ihtiyaç duymaktadır. Üreteç modeli ve verilen örnekleme göre gerçekçi üretimler yapabilen ayırıcı içeren GAN yapısını bulundurmaktadır. Pix2Pix, GAN bir giriş görüntüsüne dayanarak bir hedef görüntü oluşturuan koşullu GAN' a dayanmaktadır. Üreteç modele bir giriş görüntüsü verilmekte ve üretim gerçekleşmektedir. Üreteç ve ayırıcı modelleri, derin ağ oluşturmak için standart batchNormalization-ReLu bloklarını kullanmaktadır. Üreteç modeli için U-Net model mimarisi kullanılmaktadır. Giriş olarak bir görüntü alınır ve birkaç katman boyunca indirgeme işlemi yapılır ardından bir katmana kadar görüntü yükseltilir ve son olarak çıktı üretilir. U-Net mimarisinde görüntü indirgenir ve yükseltilir. Aşağıdaki şekilde U-Net üretici mimarisi görülmektedir.



Şekil 5. U-Net Üretici Mimarisi

Ayırıcı model, bir giriş görüntüsünü alır ve giriş görüntüsünün gerçek bir görüntü mü yoksa üretilmiş bir görüntü mü olduğunu tahmin etmekte kullanılır. Ayırıcı görüntü üzerinde evrişimli olarak çalışır ve tüm yanıtları ortalama bir biçimde alarak nihai sonucu gösterir. Aşağıda ayırıcı ağ modeli görülmektedir.

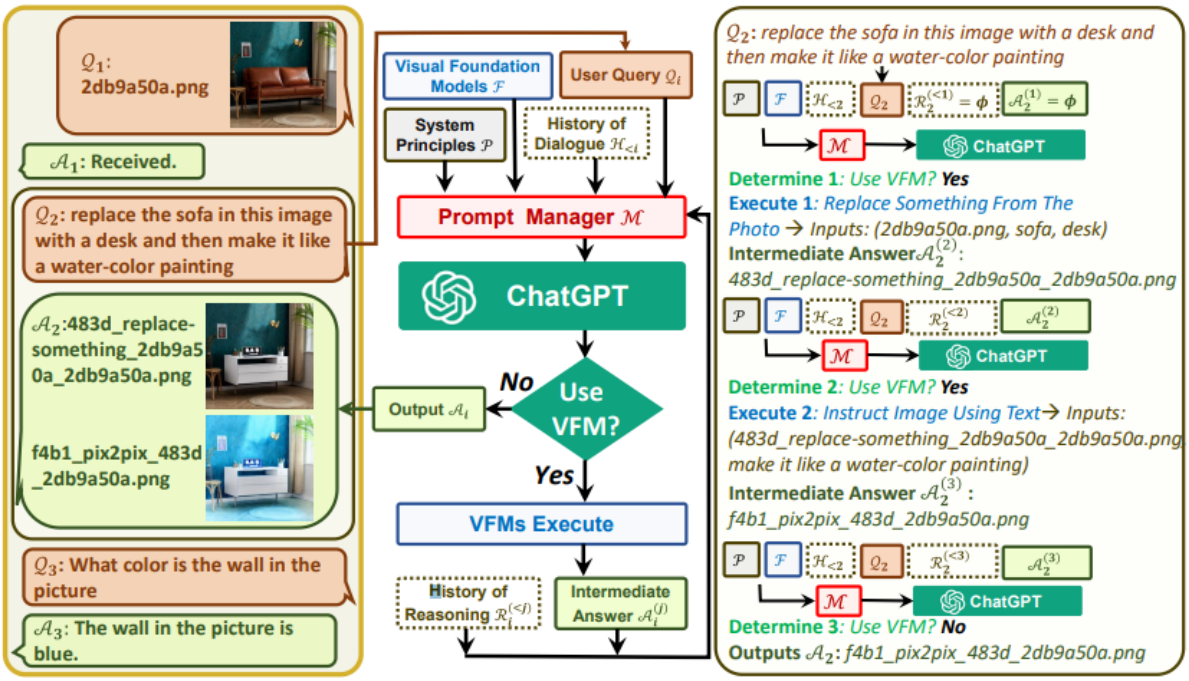


Şekil 6. Ayırıcı Ağ Modeli

ControlNet modeli ile Stable Diffusion modelleri birlikte kullanılabilir ve istenilen şekilde kontrol edilebilir. Stable Diffusion modellerinde metin komutlarıyla istenilen görüntüler oluşturmak için işlemi yönlendirmek amacıyla metin istemleri kullanılmaktadır. ControlNet ile metin istemine ek olarak bir koşul daha eklenmektedir. Örneğin Canny kenar algılayıcı kullanılarak dış hatlar saptanırsa ve algılanan kenarları içeren görüntü bir kontrol haritası olarak kaydedilir. Bu örnekte fazladan koşul olarak işlenen kenarlar ControlNet modeline giriş olmaktadı [5].

Visual ChatGPT sisteminde görüntü işlemlerinin gerçekleştirilmesi amacıyla BLIP, Stable Diffusion, Pix2Pix, ControlNet gibi modeller kullanılabilrmektedir.

Aşağıdaki şekilde Visual ChatGPT'nin genel bir bakışı görülmektedir.



Şekil 7. Visual ChatGPT Genel Bakışı

Şekil 2' de sol tarafta üç aşamalı bir diyalog gösterilmektedir. Orta tarafta, Visual ChatGPT'nin tekrarlı bir şekilde Görsel Temel Modelleri (VFMs) nasıl çağırdığını ve cevapları ürettiğini gösteren bir akış diyagramı gösterilmektedir. Sağ tarafta ise ikinci sorunun detaylı süreci gösterilmektedir. Ayrı ayrı üç bölüm aşağıdaki paragraflarda incelenmiştir.

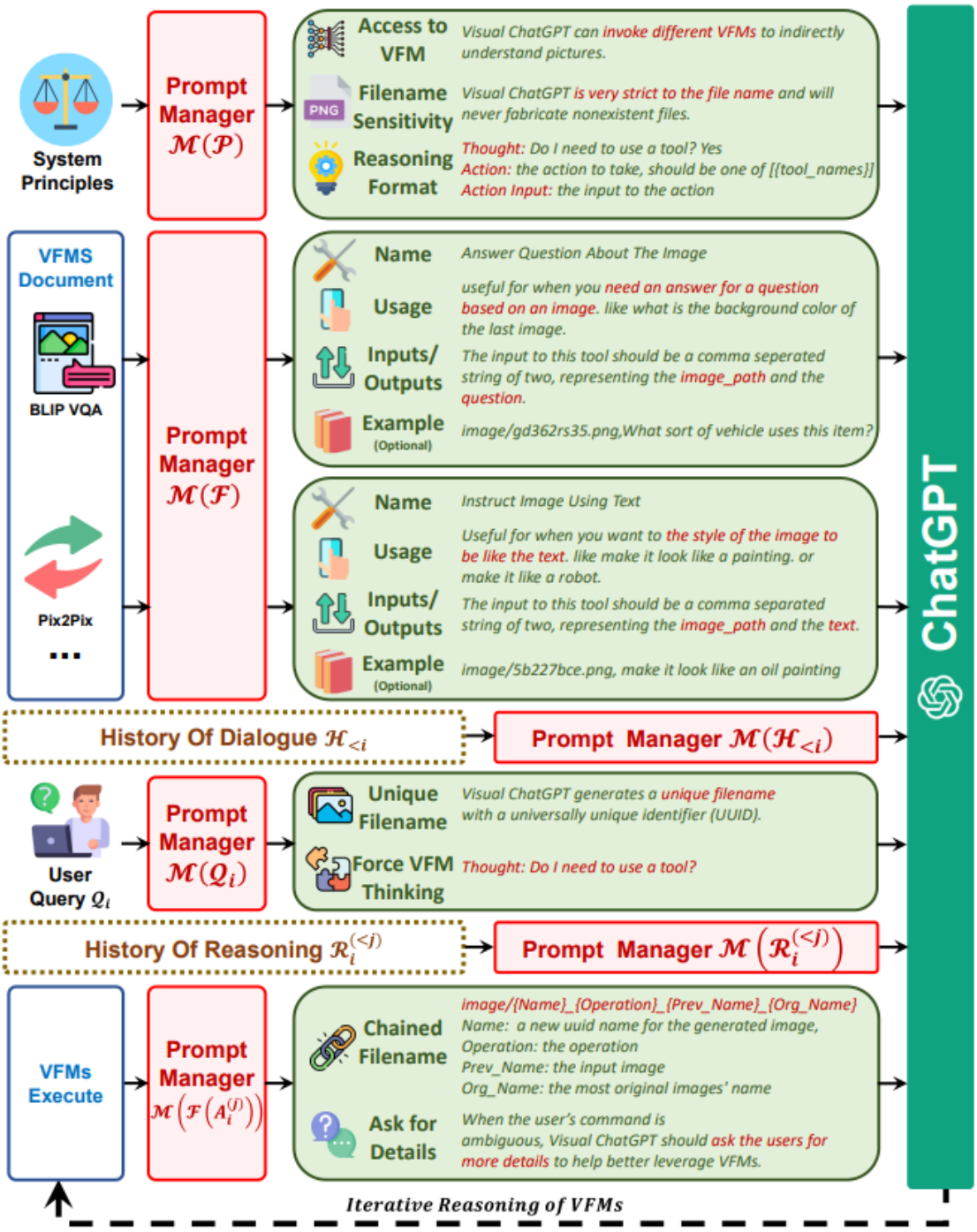
Sol kısımda Q1 ilk sorusunda kullanıcı bir görsel yüklemektedir ve Visual ChatGPT tarafından A1 cevabı ile "Received" yani görsel teslim cevabı verilmektedir. Q2 ikinci sorusunda görseldeki koltuğun bir masa ile değiştirilmesi ve görsel yağlı boya tarzında olması istenmektedir. A2 cevabı ile Q2 sorusundaki istek yerine getirilmektedir. Q3 te ise görseldeki duvarın rengi sorulmakta ve A3 ile cevabı alınmaktadır.

Orta kısımda Görsel Temel Modellerden (VFMs) bahsedersek, User Query yani kullanıcı sorgusu kullanıcının sisteme girdiği sorulardır. VFM' ler sistem prensiplerine bağlı olarak prompt menager ile iletişim halindedir. Prompt manager, ChatGPT ile entegre olarak VFM kullanılıp kullanılmamasına karar vermektedir. Eğer VFM' e gerek yok ise A çıkışını vermektedir. Ancak VFM kullanılacaksa kullanıcıya gösterilmeyen ara cevaplar üretir ve prompt menager' e işlem geri döner. Ne zaman VFM kullanılmasına gerek olunmaz ise kullanıcıya nihai cevap gösterilir. Kullanıcıya gösterilmeyen ara cevaplar ile nihai cevabın geliştirilmesi amaçlanmaktadır.

Sağ kısımda ise Q2 ikinci sorusunu incelersek; kullanıcı görseldeki koltuğu bir masa ile değiştirilmesini ve yağlı boya tarzında olması istemektedir. Prompt manager bu durumda VFM kullanılmasına karar verir ve kullanıcıya gösterilmeyen kendi içerisinde değerlendirilmek üzere bir ara cevap üretir. Bu ara cevapta ilk olarak koltuğun yerine masa konulmaktadır. Ardından akış devam eder ve VFM kullanması gerektiğine yine karar verir ve görseli yağlı boya tarzına çevirir ve yine kendi içerisinde ara cevap üretir. Döngü içerisinde VFM kullanması gerekmediğine karar vererek nihai sonuç olan görseli kullanıcıya gösterir.

7. VISUAL ChatGPT

Visual ChatGPT çıkışlarını kurallar, görsel temel modeller (VFMs), diyalog geçmişi, kullanıcı sorgusu (user query) akıl yürütme geçmişi, ara cevaplar ve prompt manager gibi faktörlere dayanarak sağlamaktadır. Şekil 3' te promp manager hakkında bir genel bakış yapılmıştır. Aşağıda Prompt Manager ile ilgili model, prensip gibi önemli bilgiler verilmiştir.



Şekil 8. Prompt Manager Genel Bakış

VisualChatGPT, görsel bilgiyi anlamlandırabilmek amacıyla çeşitli VFMs' leri birleştirerek çalışan bir sistemdir. Aşağıda VisualChatGPT' nin ilkeleri ve yönergeleri kısaca açıklanmıştır

Sistem prensiplerinin yönetimi şu şekilde açıklanabilir;

- VFMs Erişilebilirliği : Visual ChatGPT, bir VFMs listesine erişebilir ve hangilerini kullanacağına karar vermektedir.
- Dosya Adı Hassasiyeti : Belirsizlikleri önlemek amacıyla kullanılmaktadır.
- Akıl Yürütme Formatı : Visual ChatGPT, bir akıl yürütme formatını takip etmelidir bu nedenle yazar bir algoritma kullanmaktadır.
- Güvenilirlik : Yazarlar yönergeleri tasarlayarak sahte bilgiler oluşturulmasını engellemektedir.

Temel görüntü modellerinin (VFMs) yönetimi şu şekilde açıklanabilir [6];

- İsim : Her VFM için isim, VisualChatGPT' nin amacı anlamasına yardımcı olmaktadır.
- Kullanım : Belirli senaryolarda hangi VFM' in kullanılması gerektiği konusunda kararlar verebilmektedir.
- Giriş/Çıkış : Her VFM için gereken giriş ve çıkış formatlarını belirlemektedir.
- Örnek : VisualChatGPT' nin belirli bir giriş şablonu altında belirli bir VFM' yi nasıl kullanacağını ve sorguları nasıl ele alacağını anlamasına yardımcı olmaktadır.

Kullanıcı sorgusu yönetimi şu şekilde açıklanabilir;

- Visual ChatGPT, iki tür görüntüyle ilgili sorguyu ele alabilmektedir : Yeni yüklenen görüntüler için VisualChatGPT universal unique – ID (UUID) ile benzersiz bir dosya adı oluşturmaktadır ve bir ön isim eklemektedir. Bu şekilde takip edilen diyaloglar için sorgularda tekrar tekrar kullanılabilmektedir.
- Visual ChatGPT' nin VFMs' leri tetikleyebilmesi için ek bir yönerge eklenmektedir. Bu yönergede temel modellere güvenmesi ve temel modeller tarafından üretilen belirli çıktıları sağlaması teşvik edilmektedir.

Temel görüntü modeli çıktıları şu şekilde açıklanabilir;

- Visual ChatGPT, farklı VFMs' lerden gelen ara çıktıları özetleyerek sonraki etkileşimler için ChatGPT' ye iletmektedir. ChatGPT, ilerleyen işlemler için diğer VFMs' leri çağırarak devam eder ve sonlandırma koşuluna ulaşır veya kullanıcılara geri bildirim sağlar.

8. DENEYLER

Yazarlar Visual ChatGPT' nin yeteneklerini değerlendirmek amacıyla bir dizi deney gerçekleştirmektedir. Büyük dil modeli ChatGPT ile uygulanmaktadır ve Büyük dil modeli LangChain kütüphaneleri ile yönlendirilmektedir. Temel modeller HuggingFace Transformers, Maskformer ve ControlNet' ten toplamaktadır. Tüm VFM' lerin düzgün bir şekilde dağıtılması amacıyla 4 adet Nvidia V100 GPU kullanılmaktadır. Sohbet geçmişinin maksimumu uzunluğu 2000 karakterdir. Deneyler hem metin ve hem görüntü girişi yapılarak gerçekleştirilmiştir. Gerçekleştirilen çok aşamalı diyaloglarda görüntülerin tartışılması, dil ve görüntü girdilerinin sağlanması, akıl yürütme, modellerin hızlı yönetimi ve çok adımlı soruların ele alınması amaçlanmaktadır. Bu deneylerde el ile çizilen çizimlerde gerçekçi görsellere dönüştürülmesi, görsellerin derinlik bilgileriyle işlenerek anlamlandırılması, gönderilen görsellerin metin olarak tanımlanması, kenar tespit algoritmalarının kullanılması ve görselin yağlı boya gibi farklı bir tarza dönüştürülmesi gibi metin-görsel tabanlı işlemlerin başarıyla gerçekleştirildiği gözlemlenmiştir.

9. KOD İNCELEMESİ VE UYGULAMA

Bu bölümde VisualChatGPT' nin işleyişi aşama aşama kodlar üzerinden anlatılarak verilecektir ve ardından VisualChatGPT' de gerçekleştirilen uygulamalardan birkaç örnek görsel gösterilecektir.

ChatGPT yalnızca metin tabanlıdır ve görsel görevleri yapamamaktadır. Bunun önüne geçmek amacıyla kullanıcı tarafından yüklenen görüntü dosyası bir kimlik verilerek kaydedilir ve dosyanın adı ChatGPT' ye giriş olarak gönderilir ve ardından ChatGPT' den "Received" mesajı alınır. Bu işlem ChatGPT' nin sohbet geçmişinde tutularak dosyanın ileride daha sonra tekrar kullanılabilmesine imkân tanımaktadır. Aşağıdaki kod bloğu görüntü dosyası gönderme işlemi ile ilgilidir;

```
def run_image(self, image, state, txt, lang):
    image_filename = os.path.join('image', f"{str(uuid.uuid4())[:8]}.png")
    print("======>Auto Resize Image...")
    img = Image.open(image.name)
    width, height = img.size
    ratio = min(512 / width, 512 / height)
    width_new, height_new = (round(width * ratio), round(height * ratio))
    width_new = int(np.round(width_new / 64.0)) * 64
```

```
        height_new = int(np.round(height_new / 64.0)) * 64
        img = img.resize((width_new, height_new))
        img = img.convert('RGB')
        img.save(image_filename, "PNG")
        print(f"Resize image form {width}x{height} to {width_new}x{height_new}")
        description = self.models['ImageCaptioning'].inference(image_filename)
        if lang == 'Chinese':
            Human_prompt = f'\nHuman: 提供一张名为 {image_filename}的图片。它的描述是: {description}。 这些信息帮助你理解这个图像，但是你应该使用工具来完成下面的任务，而不是直接从我的描述中想象。 如果你明白了, 说 \"收到\". \n'
            AI_prompt = "收到。 "
        else:
            Human_prompt = f'\nHuman: provide a figure named {image_filename}. The description is: {description}. This information helps you to understand this image, but you should use tools to finish following tasks, rather than directly imagine from my description. If you understand, say \"Received\". \n'
            AI_prompt = "Received.  "
        self.agent.memory.buffer = self.agent.memory.buffer + Human_prompt + 'AI: ' + AI_prompt
        state = state + [(f"![](file={image_filename})*{image_filename}*", AI_prompt)]
        print(f"\nProcessed run_image, Input image: {image_filename}\nCurrent state: {state}\n"
              f"Current Memory: {self.agent.memory.buffer}")
        return state, state, f'{txt} {image_filename} '
```

Yukarıdaki kod bloğunda görüldüğü üzere sisteme bir görüntü yüklendiğinde “run_image” fonksiyonu işleme girmektedir. Bu fonksiyon “uuid” aracılığı ile yeni bir görüntü adı oluşturmaktadır ve ardından görüntü ön işleme adımları gerçekleştirmektedir. Bu işlemlerin ardından görüntü belleğe eklemektedir ve görüntü ilerleyen aşamalarda verilen id ile tekrar tekrar kullanılabilmektedir.

Görüntü dosyasının adıyla birlikte bir başlangıç girişi olarak görselin bir açıklamasına yer verilmektedir. Bu açıklama görsel modeli “BLIP” tarafından oluşturulmaktadır.

```
class ImageCaptioning:
    def __init__(self, device):
        print(f"Initializing ImageCaptioning to {device}")
        self.device = device
        self.torch_dtype = torch.float16 if 'cuda' in device else torch.float32
        self.processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
        self.model = BlipForConditionalGeneration.from_pretrained(
            "Salesforce/blip-image-captioning-base",
torch_dtype=self.torch_dtype).to(self.device)
```

ChatGPT' nin Görsel Temel Modeller (VFM)' den faydalanmasını sağlamak için yukarıdaki kod bloğunda Human_prompt değişkeninde tanımlananan açıklama ile ChatGPT' nin VFM' den yararlanmasını sağlamaktadır. Aşağıda verilen bölüm sayesinde ChatGPT' nin Görsel Temel Modeller (VFM)' den faydalanması sağlanmaktadır.

```
Human_prompt = f'\nHuman: provide a figure named {image_filename}. The description is: {description}. This information helps you to understand this image, but you should use tools to finish following tasks, rather than directly imagine from my description. If you understand, say \"Received\". \n'
```

Visual ChatGPT, bazı görsel araçları VFM modelleri ve kullanıcı arasında nasıl iletişim kuracağını belirtmek için kullanmaktadır.

Visual ChatGPT' nin hangi görsel temel modeli (VFM) değerlendireceğini tespit etmek için ajanları kullanmaktadır. Ajanlar, diğer araçlarla etkileşimde bulunmak amacıyla bir dil modelini kullanan sistemlerdir. Ajan aracılığıyla mevcut durunda kullanılabilecek tüm araçların yani görsel temel modellerin (VFMs) bir listesi sağlanmaktadır. Her bir aracın yeteneklerini detaylandırmak için bir açıklaması bulunmaktadır.

```
@prompts(name="Generate Image From User Input Text",
         description="useful when you want to generate an image from a user input text and save it to a file. "
                     "like: generate an image of an object or something, or generate an image that includes some objects. "
                     "The input to this tool should be a string, representing the text used to generate image. ")
```

Yukarıdaki kod bloğunda Araç olarak metni görüntüye dönüştüren bir Görsel Temel Model (VFM) kullanılmaktadır. Ardından ajan, aracın açıklaması ve gerçekleştirilen konuşmaları temel alaracak hangi aracı kullanacağını karar verir. Bu karar alma süreci ReAct framework kullanmaktadır. Aşağıdaki kod bloğunda ReAct framework' e yer verilmiştir.

```
self.agent = initialize_agent(
        self.tools,
        self.llm,
        agent="conversational-react-description",
        verbose=True,
        memory=self.memory,
        return_intermediate_steps=True,
        agent_kwargs={'prefix': PREFIX, 'format_instructions': FORMAT_INSTRUCTIONS,
                'suffix': SUFFIX}, )
```

ReAct, akıl yürütme paradigması olarak düşünülmektedir. Bu sayede bir akıl yürütme zinciri oluşturulmaktadır ve yanılsamaların önüne geçilmektedir. ReAct, bir dil modelinin üç adımdan oluştuğunu varsaymaktadır.

- Düşünce, akıl yürütmeyle ilgilidir.
- Eylem, ajanın yürüttüğü düşünceye dayalı bir eylem seçilmesidir.
- Gözlem, ajanın eylemlerinin sonuçlarının gözlemlendiği ve ne yapacağına karar verildiği aşamadır.

ChatGPT' nin bu üç aşamalı formatta yanıt vermesini sağlamak için aşağıdaki kod bloğunda bulunan unsurlar ChatGPT girişine entegre edilmektedir. Aşağıda araçların entegrasyonuna ait bir talimat bloğu bulunmaktadır.

Visual ChatGPT has access to the following tools:"""

VISUAL_CHATGPT_FORMAT_INSTRUCTIONS = """To use a tool, please use the following format:

```

Thought: Do I need to use a tool? Yes

Action: the action to take, should be one of [{tool_names}]

Action Input: the input to the action

Observation: the result of the action

```

When you have a response to say to the Human, or if you do not need to use a tool, you MUST use the format:

```

Thought: Do I need to use a tool? No

{ai_prefix}: [your response here]

Bu talimat bloğunda Thought, Action ve Observation adımlarının çıktıları kullanıcıya gösterilmemektedir. Tüm bilgiler kullanıcın talebini karşılayamayan yanıtlarla karşılaşmaması için gizlenmektedir. Kullanıcıya oluşturularak gösterilen tek alan ya da nihai cevap “your response here” bölümüdür.

VisualChatGPT girdileri VFM modellerine LangChain kütüphanesi araclığı ile göndermektedir. Aşağıda LangChain kütüphanesinde ilgili kod bloğu görülmektedir.

```
def _extract_tool_and_input(self, llm_output: str) -> Optional[Tuple[str, str]]:
    if f"{self.ai_prefix}:" in llm_output:
        return self.ai_prefix, llm_output.split(f"{self.ai_prefix}:")[-1].strip()
    regex = r"Action: (.*?)[\n]*Action Input: (.*)"
    match = re.search(regex, llm_output)
    if not match:
        raise ValueError(f"Could not parse LLM output: `{llm_output}`")
    action = match.group(1)
    action_input = match.group(2)
    return action.strip(), action_input.strip(" ").strip("")
```
```

Burada kullanılacak aracın ve sağlanacak girdinin çıkarılması üzerine, aracın çalıştırılması için bir çağrı yapılmaktadır.

Görsel temel modellerin (VFMs) çıktıları {Name}_{Operation}_{Prev Name}_{Org Name}. Şeklinde bir dosya adı olarak kaydedilmektedir. Bu benzersiz "uuid" işlem aracının adını temsil etmektedir. Name, benzersiz bir "uuid"' dir. Operation, aracın adını temsil etmektedir. Prev Name, yeni görüntüyü oluşturmak için kullanıcı tarafından sağlanan orijinal giriş görüntüsüdür. Bu şekilde ChatGPT yeni oluşturulan görüntü hakkında bilgi sahibi olmaktadır ve bu görüntü kullanıcıya aktarılmaktadır.

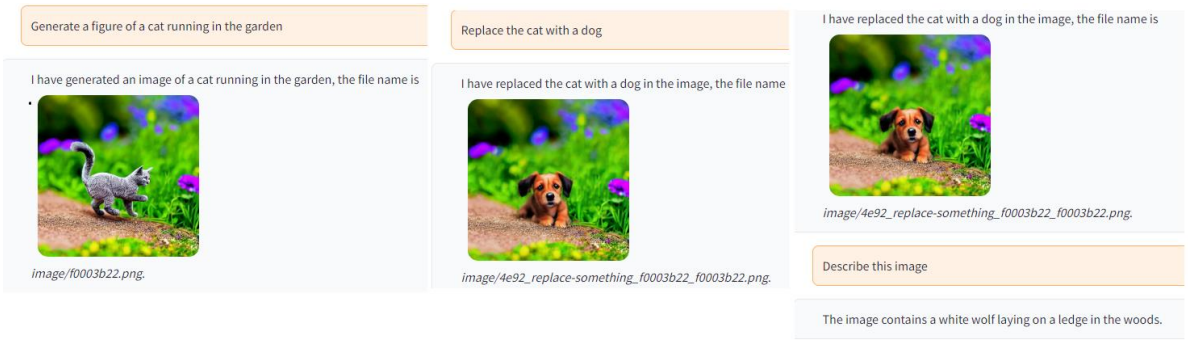
```
def get_new_image_name(org_img_name, func_name="update"):
    head_tail = os.path.split(org_img_name)
    head = head_tail[0]
    tail = head_tail[1]
    name_split = tail.split('.')[0].split('_')
    this_new_uuid = str(uuid.uuid4())[:4]
    if len(name_split) == 1:
        most_org_file_name = name_split[0]
    else:
        assert len(name_split) == 4
        most_org_file_name = name_split[3]
    recent_prev_file_name = name_split[0]
    new_file_name = f'{this_new_uuid}_{func_name}_{recent_prev_file_name}_{most_org_file_name}.png'
    return os.path.join(head, new_file_name)
```

Sonuç olarak tüm bu bileşenler bir araya getirilerek VisualChatGPT meydana gelmektedir ve metin-görsel tabanlı sohbet gerçekleştirilebilmektedir [7].

Aşağıda VisualChatGPT sistemi kullanılarak gerçekleştirilen birkaç uygulamadan görünütlere yer verilmiştir.

Visual ChatGPT' de yalnızca demo hazır komutlarında başarılı görsel sonuçlar elde edilmiş kullanıcı olarak metin ya da görsel veri girildiğinde herhangi bir sonuç alınamamıştır. Aşağıda demo hazır komutlarda elde edilen sonuçlar verilmiştir.

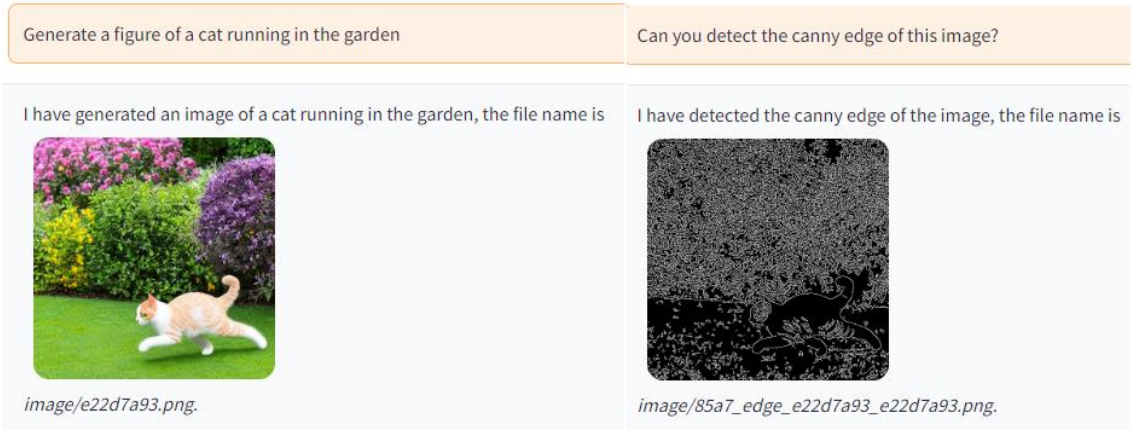
Visual ChatGPT' den bir kedi görseli yaratması istenmiştir. Kedi görseli yaratılmış ve arkaplan aynı kalarak kedi görselinin yerine bir köpek görselinin gelmesi istenmiştir. Elde edilen yeni görselin tanımlanması talimatı verilmiştir. Sonuçlar aşağıdaki şekildedir.



Şekil 9. Visual ChatGPT Verilen Talimatların Karşılaştırılması

Görselde görüldüğü gibi kedi görseli üretme ve köpek ile değiştirme konusunda başarılı olsa da görseli tanımlama konusunda biraz daha gelişmesi gerekmektedir. Köpeğin rengini ve ortamı yanlış tanımladığı görülmektedir. Bu işlemlerin gerçekleşmesi için girilen metin tabanlı komut üzerinden Stable Diffusion modeli kullanılarak bahçede koşan kedi görseli üretilmektedir. Üretilen kedi görselinin köpek görseli ile değiştirilmesinde ControlNet modeli yer almaktadır. BLIP modeli ile görselin metin olarak tanımlanması istenmektedir ve kullanıcıya metin olarak bir cevap verilmektedir.

Aşağıda Visual ChatGPT' den tekrar bir kedi görseli yaratması ve bu görselde bir görüntü işleme tekniği olan Canny Kenar Tespiti gerçekleştirmesi istenmiştir.



Şekil 10. Canny Kenar Tespiti

Şekilde gerçekleştirilen kenar tespiti görülmektedir. Nispeten başarılı olduğu söylenebilir. Şekil 10' da girilien metin tabanlı komuta göre BLIP, Stable Diffusion modelleri kullanılarak bahçede koşan kedi görseli üretilmiştir. Üretilen görsel üzerinden Canny Kenar Tespiti gerçekleştirilmek istendiğinde ControlNet modeli ek bir parametre olarak kenar değerlerini dikkate alır ve Canny Kenar Algılayıcıyı kullanarak dış hatları saptar. Algılanan kenarları içeren görüntü çıktı olarak verilir.

Visual ChatGPT gelecek vaat etmektedir ancak performans, verimlilik, etkinlik gibi konularda gelişmesi gerekmektedir.

10. SINIRLAMALAR

Visual ChatGPT, multi-modal diyaloglar (birden fazla iletişim kanalının kullanıdığı metin, görüntü vs) için umut verici bir yaklaşım olmasına rağmen bazı sınırları/sınırlamaları bulunmaktadır. Visual ChatGPT, performansını etkileyebilen ChatGPT ve görsel temel modellere bağımlıdır. İleri düzeyde prompt mühendisliği zaman alıcı olabilmektedir ve bilgisayar görüşü, doğal dil işleme gibi alanlarda uzmanlık gerektirmektedir. Belirli görevleri ele alırken sınırlı yetenekleri daha fazla görsel temel modele ihtiyaç duyulmasına yol açabilmektedir. ChatGPT' deki token uzunluğu sınırlaması kullanılabliecek görsel temel modellerin sayısını kısıtlayabilmektedir. Temel modellerin kolayca değiştirilebilmesi güvenlik ve gizlilik endişelerini düşündürebilmektedir. Ayrıca ChatGPT VFM hataları veya prompt kararsızlıkları nedeniyle tatmin edici olmayan sonuçlar üretebilmektedir.

11. SONUÇLAR VE TARTIŞMA

Bu çalışmada VisualChatGPT sistemi tanıtılmıştır. Bu sistem farklı görsel temel modelleri (VFMs) bir araya getirerek kullanıcıların metin tabanlı iletişim formatının ötesinde etkileşimde bulunmasını sağlamaktadır. Görsel verinin ChatGPT' ye aktarılması için bir dizi metot geliştirilmiştir. Visual ChatGPT' nin farklı görevlerde büyük başarı gösterdiği gözlemlenmiş ancak bazı görevlerde VFMs' in başarısız olması ya da geliştirilen metotların yetersiz kalması nedeniyle tatmin edici olmayan sonuçlarla karşılaşılmıştır. Bu sebepten ötürü yürütme sonuçlarındaki tutarlılığı kontorl etmek ve gerekli düzeltmeleri yapmak amacıyla bir öz düzeltme modülünün eklenmesi önerilmektedir. Bu modülün daha gelişmiş düşünme yeteneği kazandırması beklenmektedir. VFMs' in başarısını arttırmak için yeni modeller geliştirilebilir

Sonuç olarak tüm bileşenler bir araya getirilerek görsel bilgiyi kullanabilen Visual ChatGPT ile etkileşimli sohbet gerçekleştirilebilmektedir. Prompt manager, dosya adlarını kullanarak görsel bilgiyle yanıtları oluşturarak hangi VFM' in kullanılacağını ve VFM modellerinin çıktılarını

yönetmesine yardımcı olmaktadır. Geliştirilen bu çalışmanın ilerde daha gelişmiş görsel-metin tabanlı yapay zekâ uygulamaları için kapı açtığını söylemek mümkündür. VisualChatGPT' nin gelecek vaat etmektedir ancak performans, verimlilik, etkinlik gibi konularda gelişmesi gerekmektedir.

12. KAYNAKLAR

[1] https-1: https://medium.com/@tsaiabhi.cool/explaining-gpt-3-architecture-and-working-d0219c79202c (Erişim Tarihi:06.06.2023)

[2] Wu, C., Yin, S., Qi, W., Wang, X., Tang, Z., & Duan, N. (2023). Visual ChatGPT: Talking, Drawing and Editing with Visual Foundation Models. Microsoft Research Asia. arXiv:2303.04671v1[cs.CV].

[3] Li, J., Li, D., Xiong, C., & Hoi, S. (2022). BLIP: Bootstrapping Langıage-Image Pre-training for Unified Vision-Language Understanding and Generation. Salesforce Research. arXiv:2201.12086v2[cs.CV].

[4] https-2: https://medium.com/codex/a-quick-look-under-the-hood-of-stable-diffusion-open-source-architecture-2f07fc1e729 (Erişim Tarihi:06.06.2023)

[5] https-3: https://stable-diffusion-art.com/controlnet/#What_is_ControlNet (Erişim Tarihi:06.06.2023)

[6] https-1: https://artgor.medium.com/paper-review-visual-chatgpt-talking-drawing-and-editing-with-visual-foundation-models-e30694991e17 (Erişim Tarihi:27.05.2023)

[7] https-2: https://medium.com/mlearning-ai/visual-chatgpt-paper-and-code-review-ffe69ff16671 (Erişim Tarihi:29.05.2023)