

پروژه هوش مصنوعی

قفل های هوشمند با استفاده از سیگنال های مغزی

استاد: سرکار خانم عصایی معمم

40016341054317 یاسین رمضانی

40016341054186 آرمان طیبی

فهرست

1- استفاده از سیگنال‌های مغزی در فناوری‌های نوین

2- قفل هوشمند با استفاده از سیگنال های مغزی

3- امنیت قفل های هوشمند با سیگنال های مغزی

4- چگونگی کارکرد قفل های هوشمند

5- کاربرد های قفل های هوشمند

6- فواید استفاده از قفل های هوشمند

7- معایب استفاده از سیگنال های مغزی در قفل هوشمند

8- چالش های پیاده سازی تکنولوژی سیگنال های مغزی

9- ایده کلی

10- رندر های طرح سه بعدی

11- شبیه کد

استفاده از سیگنال‌های مغزی در فناوری‌های نوین

ماشین، و تشخیص بیماری‌های عصبی، یک -استفاده از سیگنال‌های مغزی در فناوری‌های نوین مانند کنترل بازی‌ها با تفکر، رابط‌های مغز
پیشرفته هیجان‌انگیز در علم نوروساینس و بهبود کیفیت زندگی انسان‌ها است

قفل هوشمند با استفاده از سیگنال های مغزی

روشی جذاب و پیشرفته برای نوآوری جدیدی در دسترسی به قفل ها، به وسیله استفاده از سیگنال های مغزی حفاظت از امنیت و دسترسی به فضاها و اماکن مختلف

امنیت قفل های هوشمند با سیگنال های مغزی

قفل های هوشمند با سیگنال های مغزی از طریق استفاده از الگوهای فعالیت مغز امکان کنترل و دسترسی به فضاها و اماکن را فراهم می کنند

این روش از تکنولوژی بالا و الگوریتم های پیچیده برای امنیت بیشتر استفاده می کند

چگونگی کارکرد قفل های هوشمند

قفل های هوشمند با استفاده از سیگنال های مغزی، با تشخیص الگوهای فعالیت مغز کار می کنند؛ از طریق الکترودها و سنسورهایی که بر روی سر قرار می گیرند، سیگنال های مغزی را تشخیص می دهند و با تشخیص الگوهای خاص در پردازش اطلاعات مغز قفل را باز میکند

کاربرد های قفل های هوشمند



فواید استفاده از قفل های هوشمند

امنیت بالا

با استفاده از تکنولوژی سیگنال های مغزی، قفل های هوشمند امنیت بالاتری نسبت به قفل های سنتی و رمزگذاری های معمولی دارند

راحتی و سهولت

دسترسی آسان و سریع با قفل های هوشمند به وسایل و فضاهای مختلف

فراموش نشدن رمز عبور

عدم نیاز به حفظ رمز عبور و خطر فراموشی رمز عبور در استفاده از قفل های هوشمند با استفاده از سیگنال های مغزی

پیشرفت تکنولوژی

برتری در استفاده از تکنولوژی های نوین و پیشرفته

معایب استفاده از سیگنال های مغزی در قفل هوشمند

در حال حاضر تکنولوژی مناسب برای استفاده روزمره از این ایده وجود ندارد و در صورت وجود در اوایل توسعه صرفه اقتصادی نخواهد داشت

چالش های پیاده سازی تکنولوژی سیگنال های مغزی

1

دقت تشخیص

دقت بالای تشخیص سیگنال های مغزی
در شناسایی الگوهای خاص برای
دسترسی به قفل ها

2

قابلیت تطبیق

قابلیت تطبیق با افراد مختلف و
شناسایی سیگنال های مغزی با دقت
بالا

3

رابط کاربری

طراحی رابط کاربری ساده و کاربرپسند
برای استفاده راحت و ساده از قفل های
هوشمند با سیگنال های مغزی

ایده کلی

بر پشت گوش کاربر گجتی سبک سوار میشود که با اثر انگشت کاربر روشن شده و در طول روز الگوهای مغزی را در خود ذخیره کرده و با رد شدن از آن فریم دریا نزدیک شدن به اسکنر داده توسط گیرنده از گجت دریافت میشود و در محیط ساکن به راحتی جدا شده قابل شارژ به صورت وایرلس است که هنگام خروج دوباره قابل استفاده باشد

در محیط های عمومی یا محل کار کاربر ها برای گیرنده تعریف میشوند و الگوهای کاربر های مختلف را به صورت ثبت نامی ذخیره میکند هنگام ورود دسترسی های لازم برای کاربر باز میشود و ساعت ورود و خروج به صورت خودکار ذخیره میشود



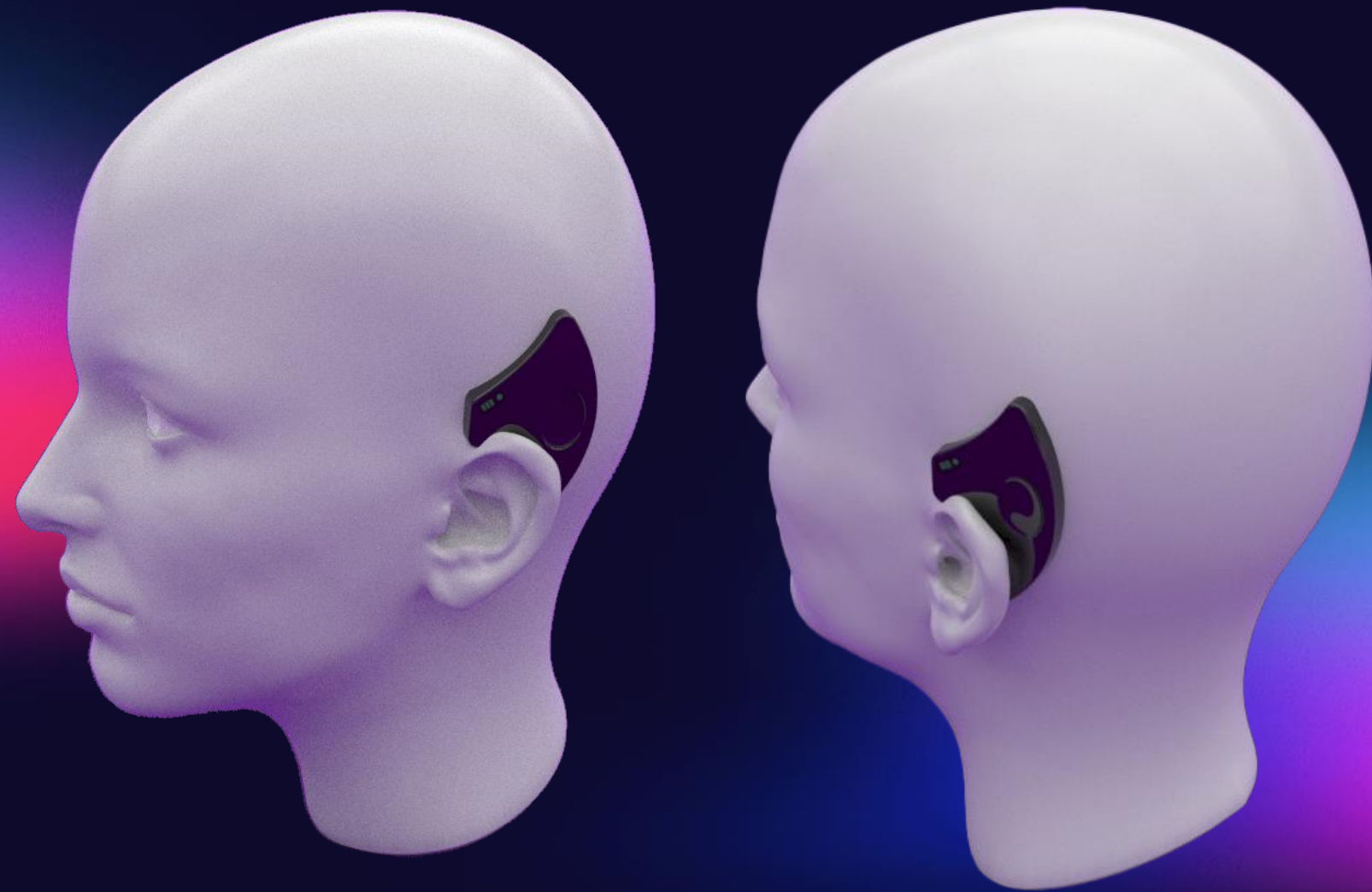
رندر سه بعدی

مدل طراحی شده در نرم افزار راینو

(فایل قرار داده شده)

رندر سه بعدی

قرار گرفتن گجت بر روی سر




```
class BrainPatternCollector:
    def init(self):
        # لیستی برای ذخیره الگوهای مغزی
        self.brain_patterns = []

    def collect_pattern(self, pattern):
        # افزودن الگوی مغزی به لیست
        self.brain_patterns.append(pattern)

class Scanner:
    def init(self, pattern_collector):
        # متغیری برای نگهداری گردآورنده الگوهای مغزی
        self.pattern_collector = pattern_collector

    def scan_brain_pattern(self, user):
        # الگوی مغزی از گجت کاربر دریافت می‌شود
        brain_pattern = user.gadget.get_brain_pattern()
        # الگوی مغزی در گردآورنده ذخیره می‌شود
        self.pattern_collector.collect_pattern(brain_pattern)
        return brain_pattern
```

```
class Gadget:
    def init(self, user_id):
        # شناسه کاربر و لیستی برای ذخیره الگوهای مغزی
        self.user_id = user_id
        self.brain_patterns = []
        # ویژگی اثر انگشت
        self.fingerprint_enabled = False

    def record_brain_pattern(self, pattern):
        # افزودن الگوی مغزی به لیست
        self.brain_patterns.append(pattern)

    def get_brain_pattern(self):
        # آخرین الگوی مغزی را باز می‌گرداند
        return self.brain_patterns[-1]

    def enable_fingerprint(self):
        # فعال کردن ویژگی اثر انگشت
        self.fingerprint_enabled = True
        print("Fingerprint authentication is now enabled.")

    def authenticate_fingerprint(self):
        # اعتبارسنجی اثر انگشت
        if self.fingerprint_enabled:
            print("Fingerprint authentication successful.")
            return True
        else:
            print("Fingerprint authentication is not enabled.")
            return False
```



```
class User:
    def init(self, user_id, gadget):
        # شناسه کاربر و گجت مرتبط با او
        self.user_id = user_id
        self.gadget = gadget

class Environment:
    def init(self):
        # دیکشنری برای ثبت نام کاربران
        self.registered_users = {}
        # گردآورنده الگوهای مغزی
        self.pattern_collector = BrainPatternCollector()
        # اسکنر برای اسکن الگوهای مغزی
        self.scanner = Scanner(self.pattern_collector)

    def register_user(self, user):
        # ثبت نام کاربر در محیط
        self.registered_users[user.user_id] = user

    def access_control(self, user_id):
        # بررسی دسترسی کاربر
        if user_id in self.registered_users:
            user = self.registered_users[user_id]
            # اسکن الگوی مغزی کاربر
            scanned_pattern = self.scanner.scan_brain_pattern(user)
            # عملیات دسترسی
            print(f"Access granted for user {user.user_id}.")
        else:
            print("User not registered.")
```

مثال استفاده:

```
smart_lock_environment = Environment()
```

ایجاد گجت و فعال کردن اثر انگشت

```
user1_gadget = Gadget(user_id="001")
```

```
user1_gadget.enable_fingerprint()
```

ایجاد کاربر و ثبت نام در محیط

```
user1 = User(user_id="001", gadget=user1_gadget)
```

```
smart_lock_environment.register_user(user1)
```

دسترسی کاربر

```
smart_lock_environment.access_control("001")
```

جدول حالات در فایل اکسل به صورت جداگانه پیوند داده شده