유튜브 동영상 자동 태깅 서비스

**경희대학교 컴퓨터공학과 2020년 1학기 캡스톤디자인1**

**팀 이름 : Profit Hunter**

**2015104192 윤영빈**

**2015104193 윤준현**

**2015104208 이태현**

**2015104209 이현규**

**요약**

2010년대 중후반부터 남녀노소 가릴 것 없이 많은 사람들로부터 이용률이 급속도로 증가하고 있는 인터넷 동영상 공유 웹사이트가 있다. 누구나 다 알고 있는 구글의 'Youtube'이다. 이 기초조사서는 Youtube 업로더가 동영상을 업로드할 때, 머신러닝을 통하여 그 동영상의 범주(카테고리, 종류, 장르 등)를 알아서 판단/구분하고 그에 적합한 태그를 달아주는 애플리케이션을 디자인 및 구현하고자 하는 내용이다.

**1. 서론**

**1.1. 연구배경**

2010년 초반 스마트폰과 무선 인터넷의 보급률이 높아지면서 가정에서는 물론, 대중교통, 카페 등 심지어 길거리에서 걸으면서까지 다양한 인터넷 동영상을 훨씬 더 간편히 시청할 수 있게 되었다. 그러자 당연한 결과로써, 동영상들을 쉽고 빠르게 검색, 접근 및 시청할 수 있는 'Youtube'(이하 유튜브) 역시 각광받고 있다.

업로드한 동영상이 높은 조회 수를 기록할수록, 그리고 동영상에 붙어 있는 광고 재생 시간과 횟수가 높을수록 동영상 업로더(이하 유튜버)는 유튜브 본사로부터 더 높은 동영상 투고 금액(수익)을 받는 시스템이다. 이 때문에 유튜버들은 단순히 재미 있는 동영상을 제작해야 하는 것뿐만 아니라, 자신들이 올린 동영상과 유튜브 페이지를 적절한 유튜브 시청자들에게 노출 및 홍보해야 한다.

또한 시청자들의 자체 영상 키워드 검색 말고도, 유튜브 쪽에서 직접 시청자들 각자가 좋아할만한 동영상의 카테고리를 분류, 파악하여 시청자들에게 영상을 추천해주는 시스템, 인터넷 유저들이 언급하는 일명 ‘유튜브 알고리즘’이라는 것이 존재한다. 업로드된 영상들의 주제와 장르를 범주화시켜서 같은 범주를 자주 보았던 다른 시청자들의 유튜브 메인 화면에 영상을 추천해주는 것인데, 유튜버들 입장에서는 이 시스템이 ‘자신의 유튜브 페이지와 동영상을 모르는 새로운 시청자들’의 시선을 끌 수 있는, 아주 중요한 시스템이라는 것이다.

이런 중요한 시스템을 앞두고 아직까지는 유튜버가 동영상을 업로드하기 전에 자신이 직접 동영상의 카테고리를 분류하고 키보드로 기입하는 방식이다. 우리는 이러한 업로드 방식을 개선하기 위해, 머신러닝을 통하여 업로드된 동영상을 읽고 파악하여 카테고리를 분류 및 추천해주는 유튜브 웹 서비스 애플리케이션을 개발한다.

**1.2. 연구목표**

유튜브 시청자들에게 업로드한 동영상을 최대한 노출시킬 수 있는 최적의 카테고리를 추천해주는 것이 궁극적인 목표이다. 다시 말해서, 이른바 유튜브 검색 엔진과 유튜브 알고리즘에 최대한 잘 적용될 수 있게 영상을 범주화시키고 태그를 달아주는 것이 핵심이라 할 수 있다.

첫 번째 목표는 머신러닝 테스트를 위한 수많은 동영상 데이터셋을 분석, 분류할 수 있는 머신러닝/딥러닝 모델을 조사하는 것이다. 영상 데이터셋으로는 사용 가능한 수백만 개의 유튜브 동영상 중 1,000개 단위의 샘플을 무작위로 선정한다. 이 샘플을 읽어 들일 수 있는 머신러닝/딥러닝 모델을 조사, 연구, 개발하여 우리가 원하는 결과를 얻어내야 한다.(비디오 분류를 통한 태그 추출)

두 번째 목표는 유튜브 웹 UI와 연동하기 위해 웹 페이지의 프론트엔드, 백엔드를 구상하고, 모델을 통하여 얻은 동영상 추천 카테고리를 영상에 꼬리표로써 다는(Video Tagging) 것이다. 1.1에서 언급했듯이, 동영상 업로드 단계 도중 유튜버 자신이 생각하는 동영상 카테고리를 기입하는 절차 대신, 잠깐의 판단하는 로딩 시간을 거쳐 유튜브 알고리즘에 잘 녹아들 수 있도록 적절한 대분류, 중분류, 소분류 태그를 추천해준다.

**2. 관련연구**

**2.1. 컴퓨터 비전**

동영상은 여러 장의 이미지들이 시간의 흐름에 따라 연속적으로 재생되는 영상물 파일이다. 따라서 동영상에 사용되는 이미지를 인식할 수 있는 기술이 요구된다. 컴퓨터 비전은 기계의 시각에 해당하는 부분을 연구하는 분야로, 이미지에서 정보를 추출한다.

**2.1.1. 물체인식**

물체인식은 미리 학습한 정보를 바탕으로, 물체를 보고 그 종류와 크기, 방향이나 위치 등 3차원적 공간정보를 알아내는 기술이다. Convolutional Neural Network(CNN)과 같은 딥러닝 모델이 사용될 수 있으며, 이를 위해 수많은 이미지들을 데이터셋으로 활용해야만 한다.

**2.1.2. 패턴인식**

패턴인식은 컴퓨터와 같이 계산이 가능한 기계장치가 어떠한 대상을 인식하는 문제를 다루는 분야로, 문자나 도형, 음성 등의 정보로부터 특징을 찾아내는 것을 말한다. 동영상과 같은 영상정보의 경우 작은 화소로 분할한 후, 각 화소의 명암을 8비트의 수치로 대치한다. 그 후 대치한 값이 급격하게 변하는 화소들을 추출하여 외곽선에 대한 정보를 얻는다. 이를 이용해 방향, 원주 등을 인식한다. 오디오 정보의 경우 강도나 위상 등에 따라 입력정보를 디지털화 하고 이룰 이용한다.

**2.2.** **머신러닝**

머신러닝 또는 기계학습은 인공지능의 한 분야로, 컴퓨터가 학습할 수 있도록 하는 알고리즘과 기술을 개발하는 분야를 말한다. 즉 기계가 코드로 명시되지 않은 동작을 데이터로부터 학습하여 새로운 데이터를 처리할 수 있도록 하는 것을 의미한다. 현재 머신러닝은 컴퓨터 과학을 포함한 대부분의 분야에서 활용되고 있는데, 컴퓨터 비전, 물체 인식 등의 분야에서 활발히 응용되고 있다.

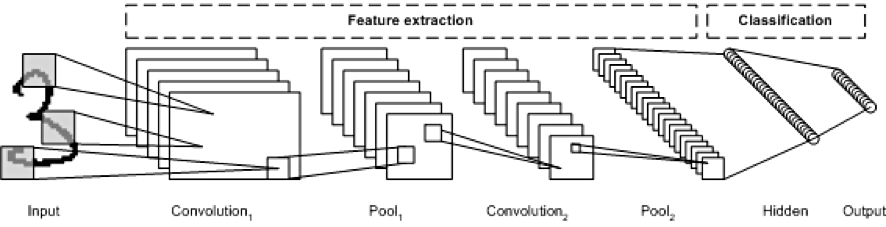
**2.2.1. 자연어 처리**

자연어 처리는 컴퓨터를 이용해 사람의 자연어를 분석하고 처리하는 기술로, 자연어를 분석, 이해, 생성하는 기술들을 사용한다. 자연어 분석은 그 정도에 따라 형태소 분석, 통사 분석, 의미 분석, 화용 분석의 네 가지로 나눌 수 있다. 자연어 이해는 컴퓨터가 자연어로 주어진 입력에 따라 동작하게 하는 기술이며, 자연어 생성은 영상이나 이미지, 표의 내용을 사람이 이해할 수 있는 자연어로 변환하는 기술이다. 영상 분석 후 가중치가 높은 태그들을 추천할 때 사용된다.

**2.3. 인공신경망 모델**

**2.3.1. Convolutional Neural Network (CNN)**

CNN(합성곱 신경망)은 이미지의 특징을 추출하여 학습하는 데 적합한 인공 신경망 모델이다. 만약 이미지 데이터로 전연결(Fully Connected) 신경망을 학습시켜야 할 경우, 3차원(공간 2차원 + RGB 1차원) 데이터를 1차원으로 평면화 시켜야 하는데, 차원을 축소하는 과정에서 필연적으로 공간 정보가 손실될 수밖에 없다. 즉 이미지를 전연결 신경망으로 학습시키는 경우에는 데이터의 특징을 잘 추출하지 못하여 결과가 좋지 못하게 된다. 이 때 이미지의 3차원 공간 정보를 손실하지 않은 상태로 학습이 가능한 모델이 CNN이다.



**[그림 1] CNN의 학습 구조 예시**

CNN의 신경망 계층은 Convolution Layer와 Pooling Layer가 반복되는 구조로 구성되어 있다. Convolution Layer에서는 합성곱 연산을 통해 이미지의 특성이 추출된 Feature Map을 생성하며, Pooling Layer에서는 Feature Map에 있는 특정 원소를 강조하거나, 중요하지 않다고 생각되는 원소를 버리는(Max Pooling) 등 최적화 연산을 수행한다. Convolution Layer와 Pooling Layer가 여러 번 반복되며 이미지의 특징을 효과적으로 추출하며, 마지막 Hidden Layer에서는 입력된 이미지가 각 category에 포함될 확률을 출력한다.

**2.3.2. Recurrent Neural Network (RNN)**

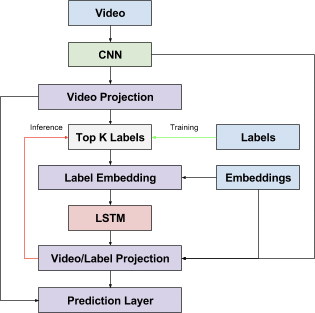
RNN(순환 신경망)은 시간 순으로 정렬할 수 있는 순차적인(sequential) 데이터를 학습하기에 적합한 인공 신경망 모델이다. 다른 인공 신경망과 다르게 내부 순환적인 전파 구조를 가지고 있어, 노드에서 출력된 값을 그대로 똑같은 노드로 다시 전파시킬 수 있다. 특유의 순환적 구조 덕에 예전에 전파되었던 데이터를 기억하는 특징을 가지고 있어, 단어의 뜻을 파악하기 위해 사전 맥락을 고려해야 하는 긴 문장의 분석 등에 유용하게 사용될 수 있다. 또한 고정된 길이의 입력으로 가변적 출력을 낼 수 있기 때문에, 출력 길이가 정해지지 않은 자연어 혹은 음성 데이터를 생성하는 문제에도 적용할 수 있다.

**2.3.3. Long Short-Term Memory (LSTM)**

LSTM(장단기 메모리)는 RNN의 Vanishing Gradient Problem을 해결하기 위해 만들어진 RNN의 변형 모델이다. RNN은 사전 정보를 기억할 수 있는 능력은 갖추고 있으나, 사전 정보와 그 정보를 사용하는 지점 사이의 거리가 멀 경우 back-propagation 시 gradient가 소실되어 제대로 업데이트가 되지 않는 문제가 발생할 수 있다. 이 문제를 해결하기 위해 LSTM은 기존 RNN 구조에서 Gate를 추가하여 연산하게 된다. Gate가 추가된 LSTM은 직전의 정보를 버릴 것인지, 만약 정보를 보존한다면 강도를 얼마나 하여 새 데이터와 결합시킬 것인지 등을 추가로 판단하게 된다. 그다지 중요하지 않다고 판단된 기존 정보는 버려지게 되므로, 중요하다고 판단되는 적절한 값만 가중치(Gradient) 업데이트 시에 효과적으로 반영되게 된다.

**2.3.4. CNN-RNN Hybrid Architecture**

효과적인 문제 해결을 위해서, 한 가지 모델만 사용하는 것보다 여러 가지 모델을 결합하여 사용하는 것이 좋은 방법이 될 수 있다. 특히, 최근 이미지 혹은 영상 분석 모델 중에서 뛰어난 성과를 거두는 결과물은 CNN과 RNN을 합성시킨 형태의 구조를 채용하는 경우가 많다. 2019년에 열린 The 3rd Youtube-8M Video Understanding Challenge에서 1위를 한 *Layer6 AI* 팀이 CNN-RNN Hybrid Architecture 구조를 채용했으며, 이외에도 대회 상위권에 입상한 팀 여럿이 해당 구조를 채용했다. 이 구조는 CNN으로 영상의 특징을 추출한 후 라벨 분류 과정에서 RNN을 사용하여, 라벨 텍스트의 의미적 해석 및 분류를 가능하게 하는 것이 특징이다.



**[그림 2] Stanford Univ, CS231n, CNN-RNN Hybrid Architecture의 예시**

**2.4. 기존 영상 자동 태깅에 관련된 연구**

현재 유튜브는 동영상 업로드 시 제목을 기반으로 태그들을 추천한다. 또한 태그가 하나 이상 입력되어 있다면 입력된 태그들 또한 다른 태그들을 추천하는 데이터로 사용한다. 즉, 영상 자체를 분석하여 태그를 추천하지는 않는다는 뜻이다.

영상 자동 태깅과 비슷한 연구로 영상 분류, 영상 인식 등이 있다. 동영상을 특정 카테고리로 분류하거나 짧은 동영상 속 상황을 자연어로 출력하는 연구들인데, 결과를 Labeling한다는 점에서 태깅과 비슷한 점이 있다. 그러나 이들의 경우 가장 가중치가 높은 결과값 하나만을 출력하고, 결과값의 논리적 상위집합과 하위집합 간의 관계를 고려하지 않으므로 다수의 키워드를 지정할 수 있는 태깅과는 다소 차이가 있다. 일례로, 자동차 집합과 승용차 집합은 논리적 관계가 존재하지만 언급한 연구들의 결과만으로 두 집합 간 논리적 관계가 있다는 것은 알기 어렵다.

현재 동영상 연구를 위해 사용되고 있는 데이터셋은 10초 내외의 짧은 길이의 영상들로, 분 단위의 영상들을 여러 조각으로 나눈 것들이다. 또한 짧은 영상 내에서도 시간 효율을 위해 모든 프레임을 이용하지 않고, 일정 간격을 둔 뒤 추출한 프레임들의 집합을 생성해 이용한다.

**2.5. 기존 연구의 문제점 및 해결 방안**

**2.5.1. 연구의 문제점**

동영상은 여러 장의 이미지들이 시간의 흐름에 따라 연속적으로 재생되는 영상 파일이지만 멀티미디어 매체이므로 영상만이 출력되는 경우는 드물고, 일반적으로 오디오와 함께 재생된다. 현재의 연구는 영상의 프레임들을 중점으로 진행되고 있는데, 오디오 정보 또한 중요하다. YouTube-8M Challenge에서 상위를 차지한 팀들은 영상과 오디오 정보를 모두 이용하는 모델을 사용했다.

그러나 단지 오디오 정보를 사용한다고 모델의 성능이 좋게 나오는 것은 아니다. 하나의 영상에 여러 주제가 담겨 있거나 여러 항목이 존재하는 데이터의 경우, 또는 영상은 움직이지 않지만 오디오만 재생되는 데이터의 경우와 같이 예외가 존재한다. 또한 모델이 분석하지 못할 정도로 짧은 효과음, 폭발음과 같이 모델의 성능을 저하시키는 경우도 있다.

**2.5.2. 해결 방안**

**2.5.2.1. 백엔드 동영상 분할**

동영상 연구를 위한 데이터셋의 경우 10초 내외의 짧은 영상들로 이루어져 있고, 하나의 주제 및 키워드가 할당되어 있다. 그러나 사용자가 유튜브에 실제 업로드하는 영상들은 수 분에 이르는데, 백엔드에서 입력 영상을 짧게 분할해 사용함으로써 앞서 언급한 몇 가지 예외사항들 중 영상에 여러 주제가 담겨 있는 경우를 줄일 수 있다.

또는 비슷한 방법으로 입력 영상 전체에서 일정 간격으로 추출한 프레임들을 하나가 아닌 여러 개의 집합으로 나눔으로써 효율을 높일 수 있을 것이다.

**2.5.2.2. 다중라벨 분류**

앞서 언급했듯 기존의 연구들은 결과가 되는 원소나 집합이 논리적으로 어떤 집합에 속해 있을 때, 결과를 출력하기 위해 논리적 관계를 고려하지는 않았다. 2018 YouTube-8M challenge에 사용되었던 Chaining기반의 다중라벨 분류 알고리즘이 본 문제를 해결하는데 도움을 줄 수 있을 것이다. 이는 논리적으로 큰 집합의 원소부터 작은 집합의 원소를 차례로 예측하는 알고리즘이다. Chaning기반의 다중라벨 분류 알고리즘을 이용한다면 가중치가 높은 값들이 서로 논리적 관계를 가지고 있을 가능성이 높으므로, 가중치가 높은 상위 몇 개의 결과들을 추천 태그로 활용할 수 있을 것이다.

**3. 프로젝트 내용**

**3.1. 서비스 소개**

**3.1.1. 기본 화면**

**스크린샷이(가) 표시된 사진

자동 생성된 설명 스크린샷이(가) 표시된 사진

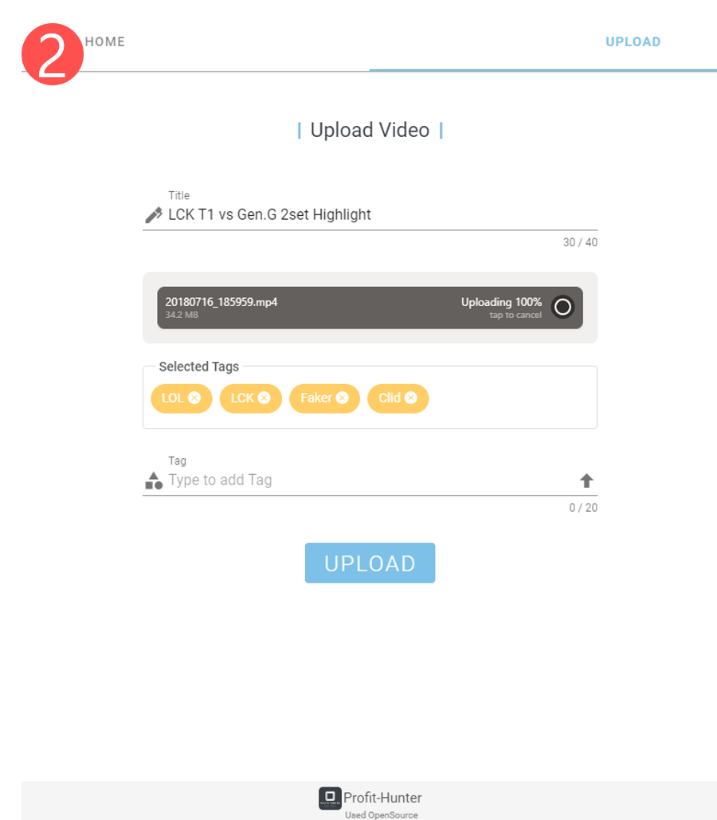
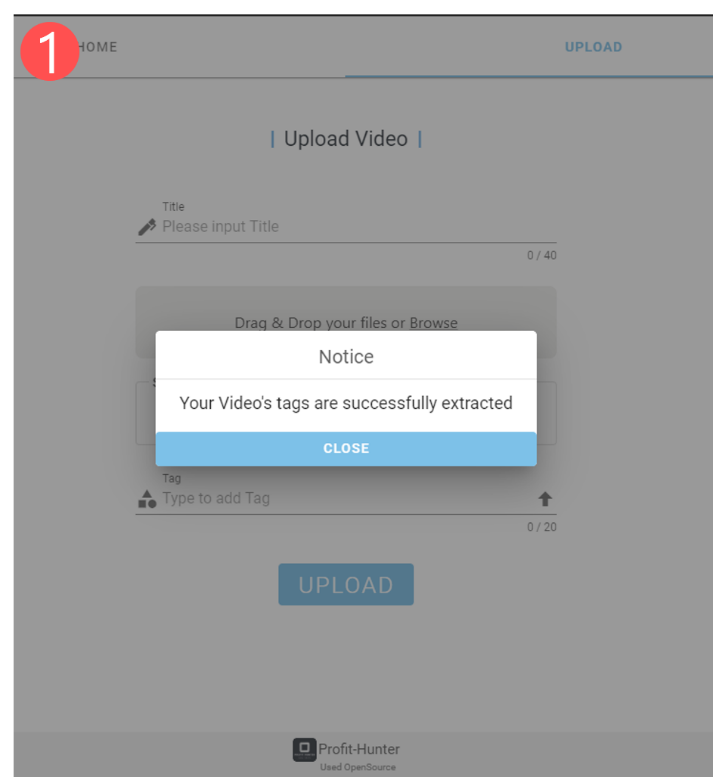
자동 생성된 설명**

**[그림 3] 홈 화면 [그림 4] 업로드 화면**

**[그림3]**은 사이트에 처음 접속했을 시 나오는 홈 화면이며, 현재 업로드되어 있는 동영상 목록을 보여준다. 1번 항목은 동영상을 태그로 검색하는 부분이며, 업로드 되어 있는 모든 태그가 리스트에 출력된다. 그리고 검색 창에 입력을 통해 해당 리스트를 좁힐 수 있다. 2번 항목은 실제 동영상의 썸네일 부분이다. 2번 항목을 클릭 시, 다이얼로그가 나타나 동영상이 전체화면으로 재생된다. 3번 항목은 해당 동영상과 연관된 태그(업로드를 통하여 추출된 태그)가 렌더된다.

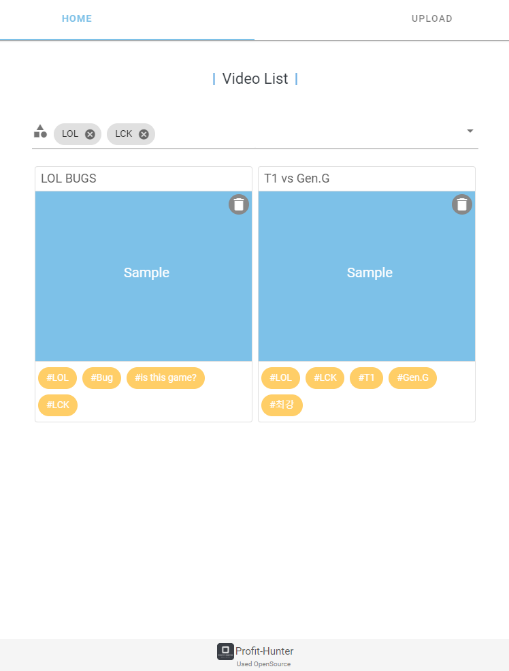
**[그림 4]**는 태그를 분석할 동영상을 업로드하는 화면이다. 1번 항목은 동영상의 제목을 입력하는 칸이다. 2번 항목은 동영상을 업로드 하는 부분이며, 클릭이나 드래그&드롭으로 동영상을 업로드 할 수 있다. 3번 항목은 태그와 연관된 부분이다. 2번에서 동영상 업로드 및 태그 분석이 완료된다면 태그가 추출되어 3번 항목에 출력시키게 된다. 추가적으로 동영상에 대한 태그를 직접 추가도 가능하다. 4번 버튼을 누르면 동영상에 대한 정보가 데이터베이스에 저장된다.

**3.1.2. 동영상 업로드 (Detail)**



동영상이 업로드되고 성공적으로 태그 분석을 마치면 ①번 사진과 같이 안내 문구가 나온다. 태그 분석을 완료하여 태그가 성공적으로 추출된 화면은 ②번 사진처럼 태그가 자동적으로 생성되며, 생성된 태그는 유저가 임의로 삭제할 수 있다. 또한 직접 타이핑하여 태그를 더 추가시킬 수 있다.

**3.1.3. 태그 검색**



홈 화면에서 태그를 검색한 화면이다. 해당하는 태그가 포함되는 동영상이 화면에 표시된다.

**3.2. 요구사항**

**3.2.1. 태그 추출을 위한 동영상 분류 학습 모델**

태그를 추출하기 위해서는 동영상을 잘 분류할 수 있는 학습된 모델이 필연적으로 구현되어야 한다. 이를 위해서 우리는 라이브러리 사용이 편리하고 오픈소스 활동이 활발한 Python라이브러리 중 하나인 ‘Pytorch’를 사용할 것이다. 구체적으로 사용할 모델은 현 단계에서는 검토 중이나, 2.3.4에서 언급했던 CNN-RNN Hybrid Architecture는 2019년 3rd Youtube 8M challange에서 1위를 차지한 모델[4]에 이용될 정도로 성능이 뛰어난 구조이기 때문에 이 서비스 또한 CNN-RNN Hybrid Architecture를 응용하여 효과적으로 태그를 추출할 수 있는 모델을 구현할 예정이다.

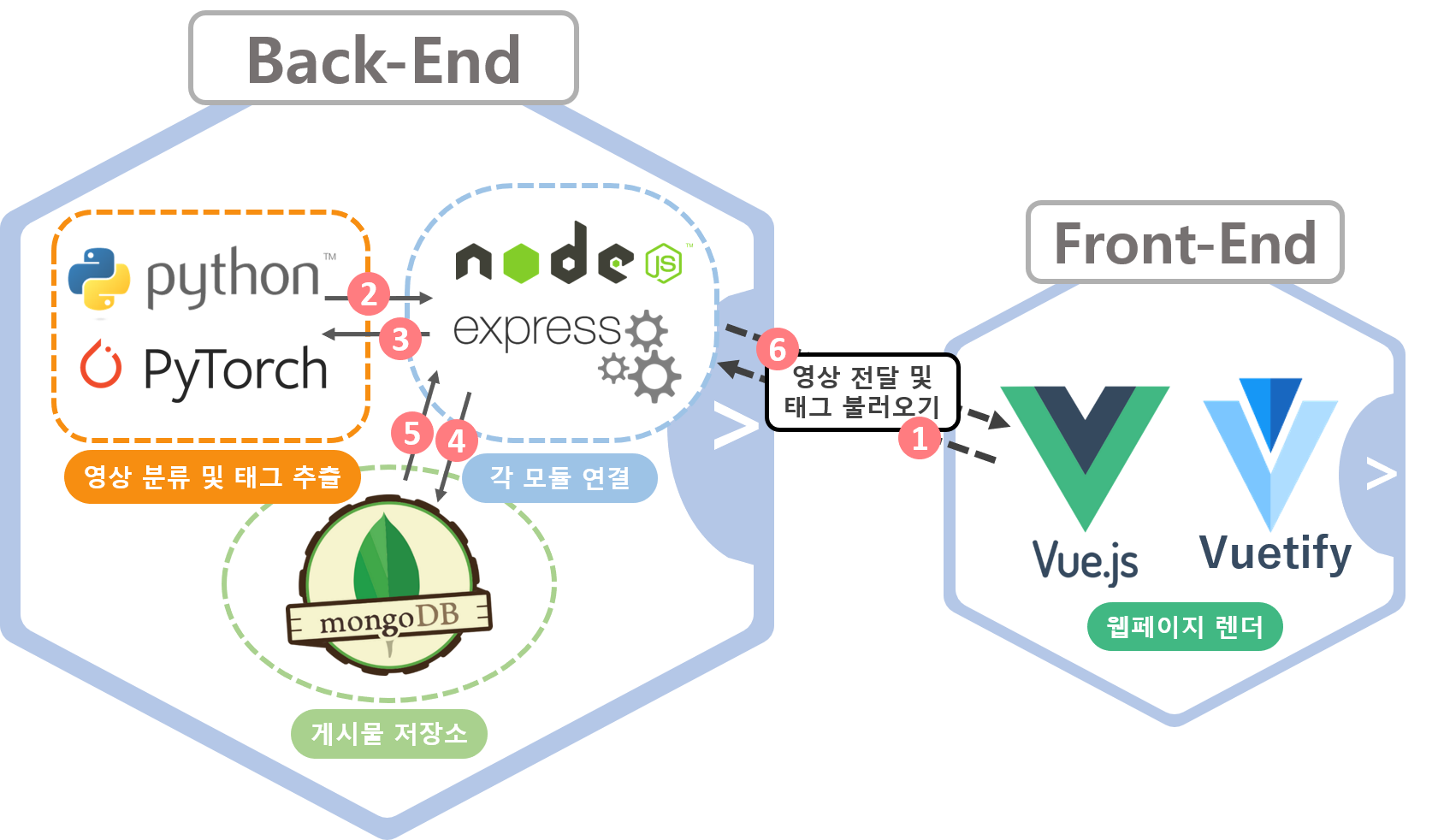
**3.2.2. 웹페이지를 위한 API서버, 프론트엔드 구현**

현재의 웹은 다양한 해상도를 가진 디바이스들이 접속하기 때문에, 웹 서비스를 만드는데 있어 반응형 웹 디자인은 필수적으로 필요한 요소 중 하나[5]이다. 우리가 사용할 프레임 워크인 Vue.js는 다른 프레임워크에 비해 가볍고, 성능 또한 좋기 때문에 현재 널리 이용 중인 반응형 웹 프레임워크이다. 그리고 Vuetify는 Google의 Material Design을 이용하기 쉽게 만들어진 Vue 전용 CSS 프레임워크이며 우리는 이 두 가지 프레임워크를 결합하여 간단하고 직관적으로 프론트엔드를 구현할 것이다. API서버는 비동기 처리로 퍼포먼스가 좋은 Node.js의 Express를 이용하여 구축할 것이며, 동영상의 정보를 간단하기 저장하기 위해 NoSQL Database인 MongoDB를 사용할 것이다. 요약하자면 이 서비스에서 사용되는 웹 페이지는 Vue.js를 이용하여 프론트엔드를, Node.js Express와 MongoDB를 활용하여 백엔드 API서버를 만들어 구현될 것이다.

**3.2.3. 동영상 분석 모델과 API서버 연결**

동영상 자동 태깅 서비스를 구현하기 위해서 최종적으로 3.2.1에서 언급한 학습 모델과 3.2.2에서 설명한 웹 페이지를 연결할 필요가 있다. 각 구조는 Python과 Javascript로 구현되어 있기 때문에 이를 효과적으로 통신할 방법이 필요하다. 이 통신에 사용될 모듈이 Node.js의 Python-Shell이다.

**3.3. 서비스 Architecture**



**[그림 5] 서비스 전체 구조**

최종적으로 우리가 구현할 서비스의 Architecture는 **[그림 5]**에서 보이는 바와 같다. 에서 보이는 바와 같이 프론트엔드에서 사용자에게 동영상을 입력 받아 API서버로 보낸다. 와 같이 Node.js에서 Python-Shell을 통해 Python으로 작성된 동영상 분류 모델을 실행시켜 태그를 추출하여 그 결과를 과 같이 다시 Node.js로 보낸다. 에서 해당 데이터들을 MongoDB에 저장하고 에서 데이터베이스에 저장된 동영상 정보를 다시 웹에 전송시킨다.

**4. 향후 일정 및 역할 분담**

**4.1. 향후 일정표**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 구분  활동 | 캡스톤디자인1 향후 추진 일정(week) | | | | | | | | | | | | | | | 비고 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 팀 구성 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 문제인식 및 문제정의 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 요구사항 분석 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 프로젝트 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 프로그램 구현 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 테스트 및 디버그 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 최종 보고서 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 합계 | 10% | | | | 40% | | | | 30% | | | | 20% | | | 100% |

**4.2. 역할 분담**

윤영빈 : 비디오 분류 모델 분석 및 구현

윤준현 : 웹페이지 구현

이태현 : 보고서 작성, 논문 분석

이현규 : 비디오 분류 모델 분석 및 구현

**5. 결론 및 기대 효과**

2020년대로 들어선 요즈음에도 유튜브 이용률은 꾸준히 증가하고 있다. 작년 19년 초등학생 대상 희망직업 조사 통계에서 크리에이터(유튜버)가 운동선수, 교사에 이어 3위를 차지할 정도로 너도나도 할 것 없이 유튜브를 많이 애용하는 상황이다.

우리가 개발하고자 하는 이 웹 서비스의 이름을 우리는 ‘태그 머니?’(Tag money?)라고 부를 것이다. 이러한 사회 속에서 ‘태그 머니?’는 유튜브를 통하여 돈을 벌어야 하는 유튜버에게 매우 유익할 것이다. 올린 영상을 빠르게 분석하고, 유튜브 검색 기능과 유튜브 알고리즘을 위한 적절한 태그를 달아준다면 동영상의 조회 수는 이전보다 더 빠른 속도로 늘어날 것이다.

유익한 것은 유튜버 뿐만이 아니다. 맞는 취향의 시청자들에게 더욱 잘 추천해준다는 것은 즉, 좋아할만한 영상을 더 쉽고 정확하게 접근할 수 있는 시청자 본인들에게도 당연히 좋은 것이다. 이를 바탕으로 유튜브 이용률이 올라간다면 광고 수익을 벌 수 있는 구글과 유튜브 본사 역시 크나큰 이득이다.

캡스톤디자인1의 우리 팀 이름인 “Profit Hunter"처럼, 우리 모두가 이윤을 추구할 수 있는 바로 이 ‘태그 머니?’(Tag money?)가 이러한 이점들을 누릴 수 있도록 도와줄 것이다.

**6. 참고 문헌**

[1] CNN, Convolutional Neural Network 요약 (<http://taewan.kim/post/cnn/>)

[2] LSTM(RNN) 소개 (<https://brunch.co.kr/@chris-song/9>)

[3] CS231n, Deep Learning Youtube Video Tags

(<http://cs231n.stanford.edu/reports/2017/pdfs/28.pdf>)

[4] CCRL (<https://arxiv.org/pdf/1911.08548.pdf>)

[5] 반응형 웹(<https://www.samsungsds.com/global/ko/support/insights/Responsive_web_1.html>)