

POLITECNICO DI MILANO

Corso di Laurea Magistrale in Ingegneria Aeronautica

SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

Dipartimento di Scienze e Tecnologie Aerospaziali



Machine learning nell'aerodinamica di una vettura di Formula 1

Relatore: Prof.ssa Antonella Abbà

Correlatore: Dott. Ing. Ben Agathangelou

Correlatore: Dott. Ing. Arron Melvin

Correlatore: Dott. Ing. Davide Paganelli

Tesi di Laurea di:

Antonio Abbate, matricola 883265

Anno Accademico 2019-2020

Ringraziamenti



Figura 1: Monoposto di Haas F1 Team con Mick Schumacher al volante durante i test in Bahrain nel 2021

Un sentito ringraziamento alla Prof.ssa Antonella Abbà, al Dott. Ing. Ben Agathangelou, al Dott. Ing. Arron Melvin, al Dott. Ing. Davide Paganelli ed a tutto il team della Haas per avermi supportato in questo lavoro di tesi.

"Adstricti sunt in continendo patrimonio, simul ad iacturam temporis ventum est, profusissimi in eo cuius unius honesta avaritia est [...]"

"Sono avari nel tutelare il patrimonio, ma, quando si tratta del proprio tempo, sono generosissimi dell'unica cosa di cui sarebbe bene essere avari [...]"

[Seneca - De brevitae vitae]

Sommario

Nel mondo della Formula 1 l'innovazione è da sempre un punto cardine: ogni squadra è sempre alla ricerca di qualcosa di rivoluzionario ed in grado di fare la differenza in pista e non solo. Il *machine learning* è ormai utilizzato in ogni settore e negli ultimi anni è entrato anche nel mondo della Formula 1. Questa tesi tratta la creazione di un programma di *machine learning* scritto in *Python* come supporto agli aerodinamici. Una parte del lavoro degli ingegneri di ogni scuderia consiste nell'analizzare e confrontare le linee di sviluppo della propria monoposto rispetto a quelle avversarie attraverso lo studio dettagliato di innumerevoli foto a loro disposizione. L'obiettivo è stato quello di fornire agli aerodinamici le immagini già categorizzate e consultabili nel minor tempo possibile selezionando semplicemente degli appositi filtri. Per questo si è creato un programma in grado di etichettare autonomamente le foto in poco tempo. Dato il largo numero di dati da processare si è inclusa anche una funzione in grado di eliminare eventuali immagini simili o a bassa risoluzione. Il passaggio fondamentale di questo lavoro è stata l'impostazione dei corretti parametri del programma di *machine learning*. Ciò ha permesso di raggiungere un'ottima percentuale di precisione (80%: 4 immagini su 5 sono classificate correttamente, l'immagine restante o è senza etichetta o è con una sbagliata). Nonostante questo margine di errore il programma sviluppato ha permesso di velocizzare notevolmente questa parte del lavoro degli aerodinamici.

Abstract

Innovation has always been a key point in the Formula 1's world, every team is always seeking for something revolutionary to make the difference not only on the track. Machine learning is now used in every sector and in the last years it also entered the Formula 1's world. This thesis is about the creation of a machine learning program, written in Python, as a support to the aerodynamicists. Part of the work of the engineers in each team consist in the analysis and compare of the development lines of their car against the opponents' ones. To do so they study in detail the countless photos available to them. The goal was to provide to the aerodynamicists the pictures already tagged and available in the shortest time possible by just selecting the appropriate filters. This is the reason why a program, that can autonomously tag photos in a short time, has been created. Given the large number of data to be processed, has also been included a function that automatically skip similar or low-resolution pictures. The fundamental step of this project was the setting of the correct parameters of the machine learning program. This made it possible to achieve an excellent percentage of accuracy (80%: 4 out of 5 images are classified correctly, whilst the remaining one is either without a label or with a wrong one). Despite this margin of error, the developed program made it possible to significantly speed up this task of the aerodynamicists.

Indice

Elenco delle figure	8
Elenco delle tabelle	10
Introduzione	12
1 Il Machine learning e i suoi campi applicativi	16
1.1 Definizione di Machine Learning	16
1.1.1 Cenni storici	19
1.2 Maggiori ambiti applicativi	20
1.2.1 Riconoscimento di immagini	21
1.2.2 Il machine learning nella medicina	22
1.2.3 Il machine learning nella geologia	23
1.2.4 Il machine learning nell'automotive	23
2 Il machine learning come supporto dell'aerodinamica in F1	25
2.1 L'Aerodinamica	26
2.1.1 Cenni storici di aerodinamica	26
2.1.2 Principi fondamentali di aerodinamica	31
2.1.3 Aerodinamica e CFD	34
2.2 L'aerodinamica e la F1	39
2.3 Il machine learning a supporto degli aerodinamici	47
3 Il software elaborato	49
3.1 Il modello	51

3.1.1	Python	51
3.1.2	Rete neurale convoluzionale	52
3.1.3	La struttura del modello ed esempio applicativo	55
3.2	Problematiche affrontate	59
3.2.1	Overfitting	59
3.2.2	Velocità di esecuzione	60
3.2.3	Escludere le immagini uguali dal processo di etichettatura	61
3.2.4	Conservazione delle immagini e loro visualizzazione . . .	62
3.3	Confronto tra alcuni risultati ottenuti	62
4	Conclusioni	65
4.1	Miglioramenti nel programma elaborato	65
4.1.1	Utilizzo di un software diverso da Sharepoint	66
4.1.2	Migliorare la precisione del modello	66
4.2	Il machine learning e la CFD	66

Elenco delle figure

1	Monoposto di Haas F1 Team con Mick Shumacher al volante durante i test in Bahrain nel 2021	3
2	Confronto di ale frontali di diverse monoposto di F1	15
1.1	Tipi di apprendimento nel <i>machine learning</i> . L'apprendimento può essere <i>Supervised</i> , <i>Unsupervised</i> , <i>Semisupervised</i> o alternativamente <i>Reinforcement learning</i> . L'apprendimento supervisionato può essere ulteriormente raggruppato in classificazione e regressione in base all'obiettivo, se categorico o continuo.	17
1.2	Principali algoritmi di <i>machine learning</i> e i maggiori ambiti applicativi	21
2.1	Esempio di analisi dati tramite il servizio di <i>machine learning</i> di Amazon	25
2.2	Tubo di Pitot	27
2.3	La prima galleria di Crocco	30
2.4	Forze che agiscono sul velivolo	31
2.5	Mach critico di un profilo	33
2.6	Auto con forze aerodinamiche indirette	41
2.7	Carico aerodinamico e resistenza delle componenti principali nel 2009	42
2.8	Coefficiente di pressione: a vista dall'alto, b vista dal basso	44
2.9	andamento del coefficiente di pressione	45
2.10	Iso-contour con $Q = 50.000l/s^2$. a) vista dall'alto, b) vista dal basso	45

2.11	vortici assiali. a) ala frontale, b) bargeboard, c) sotto l'ala anteriore , d) diffusore	46
2.12	Andamento della vorticità assiale	46
3.1	Rappresentazione di una possibile rete neurale convoluzionale con 3 livelli di convoluzione e 3 di pooling	53
3.2	Esempio di un filtro e di un immagine "caratterizzata" dopo l'operazione di convoluzione	54
3.3	Filtri applicati: team Haas durante i Test di Barcellona del 2020	57
3.4	Filtri applicati: tutte le immagini di auto viste da davanti . . .	58
3.5	Overfitting: la linea verde rappresenta un modello con overfitting, la linea nera rappresenta un modello regolarizzato. La linea verde segue meglio i dati ma ne è troppo dipendente e si rischia di avere un errore molto più alto sui nuovi dati rispetto alla linea nera	60
3.6	Esempio di visualizzazione con etichettatura effettuata a tutte le immagini	61
3.7	Esempio di visualizzazione con il filtro delle immagini uguali . .	61

Elenco delle tabelle

2.1	Contributo di deportanza delle singole componenti del veicolo .	43
2.2	Contributo della resistenza delle singole componenti del veicolo	43
3.1	Risultati senza e con i livelli di dropout	63

Introduzione

L'apprendimento automatico, o *machine learning*, è il campo dell'informatica che fornisce ai computer la capacità di imparare ad eseguire un compito senza essere stati esplicitamente programmati per la sua esecuzione. Evolutosi dagli studi sul riconoscimento di modelli e sull'apprendimento computazionale teorico nel campo dell'intelligenza artificiale, il *machine learning* esplora lo studio e la costruzione di algoritmi che permettono l'apprendimento di informazioni a partire da dati disponibili e forniscono la capacità di predire nuove informazioni alla luce di quelle apprese.

Una più formale ed ampiamente citata definizione per il *machine learning* è stata formulata da Tom M. Mitchell: "*Si dice che un programma per computer impara dall'esperienza E nei confronti di una certa classe di compiti C e insieme di misure di performance P se le sue performance sui compiti di C , secondo le misure definite in P , migliorano grazie all'esperienza E .*" [1]. Questa definizione è rilevante poichè formula il *machine learning* in termini fondamentalmente operazionali piuttosto che cognitivi, e dunque segue la proposta formulata da Alan Turing nel suo articolo "Computing Machinery and Intelligence" [2] secondo la quale la domanda "Le macchine possono pensare?" risulta poco appropriata e dovrebbe essere sostituita da "Possono le macchine fare quello che noi (come esseri pensanti) possiamo fare?".

Da ciò si deduce che l'aspetto più importante del *machine learning* è la ripetitività, infatti i computer imparano da elaborazioni precedenti per produrre risultati e prendere decisioni che siano affidabili e replicabili. In sostanza, gli algoritmi di machine learning usano metodi matematico-computazionali per apprendere informazioni direttamente dai dati, senza modelli matematici ed

equazioni predeterminate, e migliorano le loro prestazioni in modo "adattivo" all'aumentare degli "esempi" da cui apprendere.

Il *machine learning* ha ormai conquistato ogni settore, non poteva mancare nella Formula 1.

Negli ultimi anni la vettura più veloce in pista è sempre stata, sostanzialmente, quella con la migliore aerodinamica e con il miglior bilancio tra resistenza e deportanza. Progettare quindi un'auto estremamente efficiente da questo punto di vista è fondamentale per ottenere risultati migliori. Il lavoro degli aerodinamici non è certo semplice, bisogna tener conto di molteplici fattori e di molti dati da analizzare nel minor tempo possibile.

In aiuto agli aerodinamici c'è il lavoro svolto dai fotografi. Le loro foto scattate durante le prove o le gare, vengono analizzate dettagliatamente e consentono di poter fare un'analisi sulle linee di sviluppo delle varie monoposto. Innumerevoli sono le immagini e molteplici sono i dati da registrare.

È per questo che nel team di Formula 1 Haas, è stata portata avanti l'elaborazione di un software di *machine learning* per supportare il delicato, ma fondamentale compito degli aerodinamici. Questo lavoro ha permesso di mettere in luce come la giusta programmazione possa essere un supporto valido per chi svolge delle mansioni di elaborazione con tanti dati.

È proprio a valle di ciò che si intuisce come il ruolo dell'aerodinamico in grado di programmare e costruire programmi di *machine learning* diventerà fondamentale nell'immediato futuro. Se il programma "impara dall'esperienza" e compie azioni "operazionali e non cognitive" allora chi lo programma, dovrà essere un valido maestro in grado di dargli le basi per poi compiere le operazioni necessarie. Un team informatico privo di un supporto aerodinamico potrebbe non avere le basi tecniche e nozionistiche per rendere il programma funzionale e al servizio dell'aerodinamica.

Ad oggi non esiste un'ampia bibliografia in merito all'utilizzo del *machine learning* in ambito aerodinamico o delle competizioni di F1, ma ci si è appoggiati ai numerosi ambiti applicativi della medicina e di altre branche della scienza

per poter utilizzare il riconoscimento immagini anche nell'ambito di studio di questa tesi.

In questo lavoro di tesi l'ingegnere aerodinamico non solo ha messo al servizio dell'informatica le proprie competenze specifiche, ma ha programmato l'intero software "insegnandogli" le basi per compiere il proprio lavoro.

Il lavoro di tesi ha quindi l'obiettivo di spiegare come è stato creato il software che permette la categorizzazione e l'analisi di immagini di componenti delle macchine di Formula 1 al fine di andare non solo a vedere i progressi che determinate componenti hanno in relazione alle prestazioni in pista, ma anche di confrontare i propri sviluppi con quelli delle case automobilistiche concorrenti (figura 2). Come possiamo vedere dall'immagine, i profili alari che apparentemente sembrano identiche, hanno diverse filosofie di gestione dei flussi sulla parte frontale delle monoposto. Alcune squadre deviano verso l'alto la totalità dei flussi, infatti le loro ali non mostrano alcun tipo di inclinazione laterale e non presentano alcuna barriera per il flusso. L'intento è quello di far scavalcare lo pneumatico dal flusso solo con traiettorie ripiegate verso l'alto. Questa soluzione sarebbe la migliore in termini di deportanza totale. Altri team hanno deciso di deviare una parte dei flussi sui lati dello pneumatico, riducendo l'incidenza alare verso gli estremi dell'alettone. Inclinando i profili verso il bordo esterno, i tecnici hanno voluto dare la possibilità alle molecole d'aria di scivolare lungo l'apertura dell'alettone fino ad incontrare le fessure tra questo e lo pneumatico. Il flusso riuscirebbe così a infilarsi all'interno della strettoia e fuoriuscire dal lato esterno della ruota sterzante. Una scelta orientata probabilmente all'equilibrio tra deportanza e resistenza del veicolo.

Il programma di *machine learning* ideato facilita parte del lavoro degli aerodinamici confrontando, categorizzando e analizzando l'ingente numero di foto a disposizione e consentendo ai tecnici in brevissimo tempo di apportare modifiche sulle auto.

Nella prima parte della tesi si illustra come funziona il processo di apprendimento, le molteplici tecniche di machine learning e i numerosi ambiti



Figura 2: Confronto di ale frontali di diverse monoposto di F1

di applicazione dai quali si è preso spunto per raggiungere l'obiettivo della tesi.

Nel secondo capitolo si è effettuato un inquadramento del lavoro nel contesto aerodinamico a livello storico e tecnico e si è spiegato concretamente come il machine learning, e in particolare il riconoscimento e la catalogazione di immagini, possono supportare il lavoro di una squadra di Formula 1.

Nel terzo capitolo si entra nel vivo del software elaborato e si illustrano tutti i processi logici e di programmazione effettuati.

Nella quarta, ed ultima parte vengono riportate le potenzialità del lavoro, le conclusioni e gli eventuali futuri sviluppi.

Capitolo 1

Il Machine learning e i suoi campi applicativi

1.1 Definizione di Machine Learning

Il *Machine Learning* è un metodo di analisi dati che automatizza la costruzione di modelli analitici. È una branca dell'Intelligenza Artificiale e si basa sull'idea che i sistemi possono imparare dai dati, infatti riescono ad identificare modelli autonomamente e prendere decisioni con un intervento umano ridotto al minimo. Il *machine learning* nasce dall'intersezione di più settori e per questo esistono diverse definizioni di questa disciplina. Ad oggi la più accettata nel mondo della scienza è quella elaborata da Mitchell nel 1997 e già citata nell'introduzione.

L'aspetto più importante del *machine learning* è la ripetitività, perchè più i modelli sono esposti ai dati, più sono in grado di adattarsi in modo autonomo. I computer imparano da elaborazioni precedenti per produrre risultati e prendere decisioni che siano affidabili e replicabili.

Il *machine learning* si divide in 4 tipologie (figura 1.1) *Supervised*, *Unsupervised*, *Semi-supervised* e *Reinforcement Learning*.

- **Supervised:** Lo scopo qui è utilizzare un algoritmo per apprendere la funzione ottimale che cattura al meglio la relazione tra l'*input* e la variabile di destinazione. Il motivo per cui questo tipo di apprendimento è chiamato

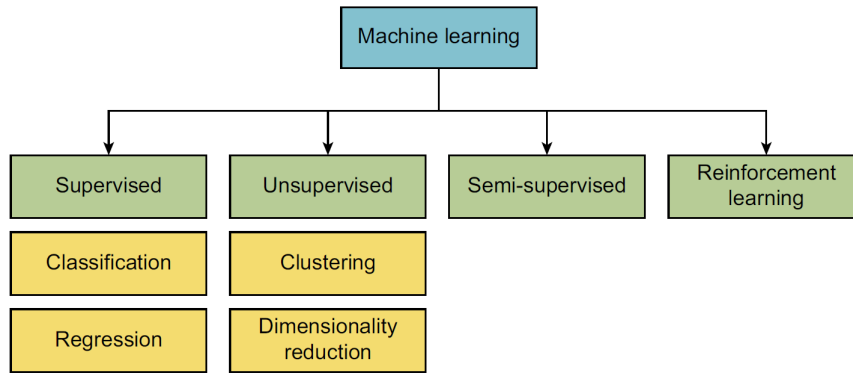


Figura 1.1: Tipi di apprendimento nel machine learning. L'apprendimento può essere *Supervised*, *Unsupervised*, *Semisupervised* o alternativamente *Reinforcement learning*. L'apprendimento supervisionato può essere ulteriormente raggruppato in classificazione e regressione in base all'obiettivo, se categorico o continuo.

"supervisionato" è che l'algoritmo ha una conoscenza preliminare di quali dovrebbero essere i valori di *output*. L'algoritmo viene addestrato utilizzando più esempi ed è autorizzato a ricevere risposte durante il processo di apprendimento in base a quanto la sua previsione corrisponde al vero obiettivo. Questo tipo di apprendimento viene spesso paragonato all'apprendimento con un insegnante: l'insegnante conosce le risposte corrette e corregge l'algoritmo quando commette un errore. In questo contesto, quindi, l'apprendimento è un processo iterativo di previsioni e successivi aggiustamenti, fino a quando la differenza tra previsioni di *output* e *target* non viene minimizzata il più possibile. Le prestazioni vengono misurate confrontando le previsioni dell'algoritmo con i valori reali in dati invisibili. A seconda che la variabile di obiettivo sia una variabile categoriale o continua, un'attività di apprendimento supervisionato è rispettivamente un problema di classificazione o di regressione.

- *Classificazione*: gli algoritmi di classificazione mirano a prevedere l'appartenenza a gruppi, noti anche come etichette o classi, per una serie di osservazioni. La risposta in *output* della macchina è quindi un dato discreto, i dati in *input* possono essere continui o discreti.
- *Regressione*: è un processo statistico che cerca di stabilire una rela-

zione tra due o più variabili. Fornito a un modello di regressione un valore x , questo restituirà il corrispondente valore y generato dall'elaborazione di x . La regressione lineare si differenzia nettamente dalla classificazione, poiché quest'ultima si limita a discriminare gli elementi in un determinato numero di classi (*label*), mentre nella prima l'*input* è un dato e il sistema ci restituisce un *output* reale (a differenza della classificazione che riceve in *input* un dato e restituisce in *output* un *label*, un'etichetta a cui il dato appartiene e in generale quindi un valore discreto)

- **Unsupervised:** è una tecnica in cui la macchina apprende dall'esperienza senza avere esempi o risposte di riferimento. I dati non sono etichettati e la loro struttura non è predefinita. A differenza dell'apprendimento supervisionato non c'è nessun valore di *target*. Questo tipo di apprendimento si divide in "clustering" e "dimensionality reduction"
 - *Clustering:* è una tecnica analitica per lo sviluppo di sottogruppi da un campione più ampio. Un problema di *clustering* è un problema di apprendimento senza supervisione che richiede al modello di trovare gruppi che presentino delle similarità tra loro.
 - *Dimensionality reduction:* è utile in situazioni in cui il numero di caratteristiche è sostanzialmente maggiore del numero di osservazioni. Richiede al modello di eliminare o combinare variabili che hanno un effetto scarso o nullo sul risultato. Questo è spesso usato in combinazione con la classificazione o la regressione.
- **Semi-Supervised:** nell'apprendimento *Semi-Supervised*, le etichette o le variabili *target* sono disponibili solo per una parte dei dati. L'apprendimento semi supervisionato affronta questo problema consentendo al modello di integrare i dati disponibili senza etichetta nel suo apprendimento supervisionato. Questo approccio è utile quando è poco pratico o troppo costoso accedere o misurare la variabile *target* per tutti i partecipanti.

- **Reinforcement learning:** l'obiettivo è costruire un sistema in grado di apprendere interagendo con l'ambiente. A differenza degli altri tre, questo algoritmo si occupa di problemi di decisioni sequenziali, in cui l'azione da compiere dipende dallo stato attuale del sistema e ne determina quello futuro. Il suo comportamento è modellato attraverso una sequenza di premi e penalità, che dipendono dal fatto che le sue decisioni verso un obiettivo definito siano corrette o errate, come definito dal ricercatore. A differenza dell'apprendimento supervisionato, in cui l'algoritmo utilizza gli esempi forniti per modellare il comportamento, nel *Reinforcement learning* l'algoritmo può comportarsi liberamente, cioè, sulla base di tentativi ed errori, scopre quali azioni massimizzano la ricompensa e minimizzano la penalità.

1.1.1 Cenni storici

L'età del *machine learning* inizia alla fine degli anni '50 con il lavoro dei pionieri Marvin Minsky [3] [4] e Frank Rosenblatt [5] con la scrittura delle prime reti neurali. Le reti neurali artificiali (*Artificial Neural Network*, ANN) sono una famiglia di modelli ispirati alle reti neurali biologiche. Esse vengono utilizzate per stimare o approssimare funzioni che possono dipendere da un numero elevato di input, molti dei quali spesso noti. Le reti neurali artificiali vengono generalmente presentate come dei sistemi di "neuroni" fra loro interconnessi, tra i quali avviene uno scambio di messaggi. I primi algoritmi però, riuscivano a compiere solo alcune operazioni elementari e non potevano portare a termine compiti complessi o fare delle generalizzazioni, questo si aggiungeva all'arretratezza delle infrastrutture computazionali di quel periodo. Tutti i codici si basavano sulla classificazione di oggetti partendo da una serie di dati di allenamento costruito in precedenza. I grandi passi in avanti che hanno migliorato notevolmente lo sviluppo di questi programmi sono stati la retropropagazione dell'errore e la convoluzione [6]. La retropropagazione dell'errore è un metodo di progettazione delle reti neurali artificiali, consistente in un meccanismo di

apprendimento, detto con supervisione, attraverso una procedura ciclica di aggiornamento della matrice dei pesi sinaptici che attenua progressivamente, al termine di ogni ciclo, l'errore fra il risultato noto e quello effettivo nel trattare una serie di casi di esempio. Le iterazioni si arrestano quando la procedura converge, ovvero quando la discrepanza scende al di sotto di un limite prefissato. Non può essere tuttavia garantito che questo avvenga in tutti i casi. La convoluzione è un'operazione matematica che va a trasformare il dato che si ha in ingresso attraverso delle operazioni matematiche tra due funzioni. Nel caso di un'immagine, la convoluzione si può intendere come un filtro che va a modificare la posizione o il valore di alcuni pixel, questo meccanismo è alla base di qualsiasi software di computer grafica. L'applicazione di queste matrici permette di mettere in risalto i particolari delle immagini che si vogliono analizzare e di ridurre il numero di neuroni che vengono trasmessi all'interno della rete neurale.

L'evoluzione delle reti neurali si è basata sullo studio delle reti neurali biologiche (il cervello); diversi concetti fondamentali del *machine learning* sono tutt'oggi pesantemente basati sulle teorie dell'apprendimento umano [7]. Quello che fa l'algoritmo non è pensare ma imparare dalle informazioni che ha per migliorare le sue prestazioni.

1.2 Maggiori ambiti applicativi

Il *machine learning* si sta sviluppando soprattutto negli ultimi anni, le risorse software e hardware sono oggi largamente fruibili da tutti anche grazie all'ampia disponibilità di software gratuiti ed open-source. Le grandi compagnie (Google, Facebook, Microsoft) hanno reso disponibili open-source i loro algoritmi di *machine learning* (Tensorflow [8], Pytorch [9], CNTK [10])

Come riportato nella figura 1.2 gli ambiti di applicazione sono molteplici e variano in base allo scopo e conseguentemente al tipo di algoritmo. Di seguito alcuni esempi di campi di applicazione.

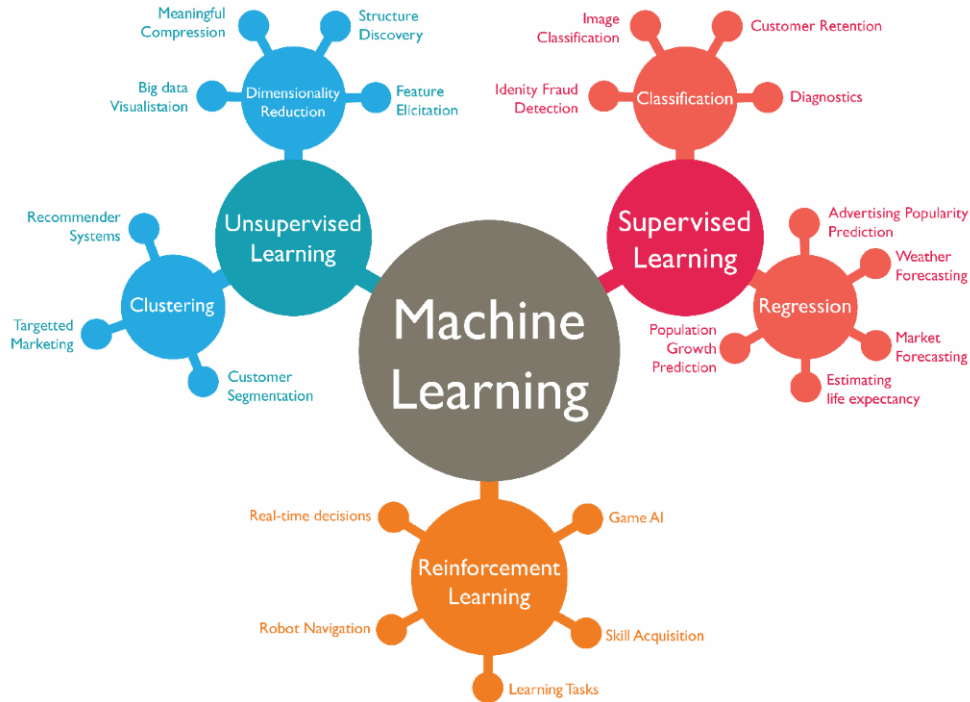


Figura 1.2: Principali algoritmi di machine learning e i maggiori ambiti applicativi

1.2.1 Riconoscimento di immagini

Sin dalla nascita della fotografia si è avuta la necessità di organizzare e archiviare le immagini. Con lo sviluppo di internet, dei database immagini e della fotografia accessibile a tutti, questa necessità non solo si è allargata, ma si è intersecata con il mondo del *machine learning*. Quest'ultimo, oltre a permettere una migliore archiviazione, ha reso l'elaborazione e l'analisi delle immagini più facilmente eseguibili. In particolare il *machine learning* dal 1990 è stato utilizzato nella archiviazione e nella ricerca automatica di immagini. È diventato con gli anni di fondamentale importanza creare algoritmi in grado di identificare e classificare le foto non solo sui dispositivi personali, ma anche nei database on line e nei motori di ricerca come Google immagini [11]. Questa necessità di analizzare e categorizzare negli ultimi anni è diventata molto importante nel mondo della ricerca e sviluppo, dalla medicina al mondo industriale. Ovviamente, come vedremo nella sezione 1.2.2 non sempre l'automatismo degli algoritmi porta a risultati positivi, ciò accade poiché gli algoritmi necessitano di continui aggiornamenti. I *software* di *machine learning* devono essere pro-

grammati e impostati da persone che non hanno solo competenze informatiche, ma che siano "studiosi" degli argomenti che l'algoritmo deve "imparare".

1.2.2 Il machine learning nella medicina

Il *machine learning* è molto utilizzato nella diagnostica medica, sia per la diagnosi che per la terapia. In particolare, per la rilevazione dei disturbi mentali [7] viene utilizzato il metodo *Supervised* e gli algoritmi di classificazione. Questi ultimi vengono usati per la ricerca dei disturbi cerebrali poiché la maggior parte dei problemi clinici può essere semplificata in una decisione categorica. Per esempio un determinato paziente deve essere trattato con i farmaci A, B o C? L'algoritmo impara a distinguere i pazienti con le varie malattie dai pazienti sani, la maggior parte degli studi di classificazione diagnostica utilizza dati di *neuroimaging* [12]. Un piccolo numero di studiosi ha utilizzato un altro tipo di dati per esempio i sintomi motori per identificare i pazienti colpiti dal Parkinson [13] e informazioni genetiche per individuare gli affetti da schizofrenia [14]. Per quanto riguarda lo studio del sistema digerente, si è usato il *machine learning* per effettuare una mappatura di tutti i referti medici. Questo ha consentito una maggiore facilità di consultazione e di diagnosi [15]. In alcuni ambiti della diagnostica medica l'utilizzo del *machine learning* non è risultato molto affidabile e ciò perchè gli sviluppatori spesso non hanno alcuna competenza medica ma anche per altre ragioni come ad esempio per l'analisi del fondo della retina. Il riconoscimento automatico delle patologie identificabili attraverso le immagini del fondo dell'occhio ha riportato un'affidabilità del 70% che è sicuramente inferiore al normale processo diagnostico fornito dai medici. Una delle possibile cause del fallimento dell'applicazione del *machine learning* in questo caso può essere la variazione elevata del fondo della retina da persona a persona, di conseguenza l'algoritmo non è in grado di distinguere tra una normale differenza e un vero problema dovuto ad una patologia [16].

1.2.3 Il machine learning nella geologia

Fin dagli anni '60 si è utilizzato il *machine learning* per descrivere la ciclicità dei depositi sedimentari, per effettuare l'analisi del sottosuolo, per l'idrologia e la vulcanologia. Negli anni successivi si è allargato il suo uso come supporto all'attività estrattiva, oceanografica e idrologica [17]. Nella geologia risulta di fondamentale importanza l'individuazione della stratigrafia e delle caratteristiche del suolo anche e soprattutto attraverso l'osservazione. Per questo attraverso l'elaborazione di milioni di immagini con il *machine learning* è facile l'identificazione di modelli geologici per classificare le caratteristiche geologiche. L'analisi e la classificazione combinate e automatizzate delle immagini tramite l'apprendimento automatico possono ridurre significativamente i tempi di analisi, i pregiudizi dell'interprete e le incongruenze. [18]

1.2.4 Il machine learning nell'automotive

Il *machine learning* non solo è un validissimo supporto al conducente nella guida del veicolo garantendo dei viaggi più sicuri, ma è anche molto diffuso nella stessa industria automobilistica. Ad esempio, viene utilizzato per valutare l'influenza dei parametri di progettazione sul peso, sui costi o sulle proprietà funzionali dei nuovi modelli di auto. Gli ingegneri automobilistici dedicano una parte considerevole del loro tempo all'analisi di queste influenze, esaminando una alla volta le simulazioni che ne derivano. Attraverso una raccolta dati e l'utilizzo del *machine learning* è possibile non solo fare una analisi strutturale del veicolo e analizzarne i punti deboli, ma andando ad incrementare l'algoritmo, anche con dati di incidenti e crash test, è possibile individuare le parti cedevoli e come agire per migliorare la struttura dell'automobile [19].

Uno degli ambiti che dimostra maggiormente i poteri e i rischi del *machine learning* sono le auto a guida autonoma [20]. Nel dipartimento di *computer science* della Brown University attraverso gli algoritmi stanno insegnando alle auto quali comportamenti non adottare attraverso filmati di autisti che infran-

gono le regole del codice della strada. Il *software*, per apprendere il grado di pericolosità di un determinato comportamento, ha “ascoltato” i commenti di un ipotetico passeggero umano che, seduto sul sedile posteriore, esprimeva il proprio parere sullo stile di guida del conducente.

Un altro settore prevede l'utilizzo dei dati GPS degli smartphone al fine di ottimizzare la scelta del percorso migliore per gli utenti. L'algoritmo identifica i percorsi preferiti e ne suggerisce di più brevi e meno trafficati [21]

Si è sfruttata invece la correlazione per creare un modello di analisi delle arterie stradali e del traffico al fine di prevedere come e dove potenziare le infrastrutture [22].

Capitolo 2

Il machine learning come supporto dell'aerodinamica in F1

In letteratura c'è una bassissima documentazione in merito all'uso del *machine learning* in F1 o in generale nel motorsport. Probabilmente questo è legato sia ad un utilizzo ancora marginale ed embrionale, sia alla riservatezza che ogni scuderia adotta per evitare di fornire idee ai concorrenti. Fino ad ora viene utilizzato principalmente come strumento per prevedere eventuali strategie e i consumi delle gomme [23].

Solo negli ultimi anni si sta assistendo ad un maggior utilizzo del *machine learning* anche in altri campi come nell'analisi della CFD o in generale nell'analisi di immagini che è argomento di questa tesi. L'utilizzo dell'apprendimento

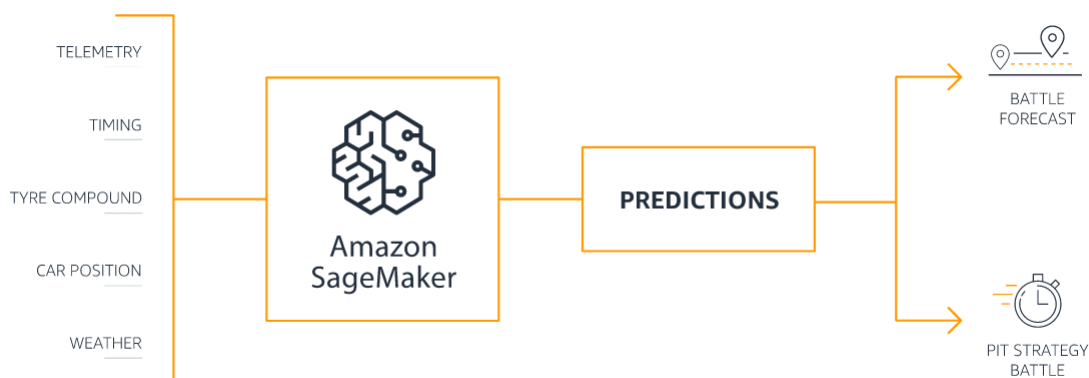


Figura 2.1: Esempio di analisi dati tramite il servizio di machine learning di Amazon

automatico diventa fondamentale poiché i dati a disposizione delle scuderie sono molti. Per ogni gran premio, ogni prova e ogni simulazione in galleria del vento ci sono centinaia di sensori che ogni secondo generano migliaia di dati di telemetria. Un'analisi puntuale di questi dati richiede lunghi tempi di elaborazione, il *machine learning* potrebbe ridurre drasticamente questi tempi andando ad effettuare un'analisi preliminare.

2.1 L'Aerodinamica

2.1.1 Cenni storici di aerodinamica

L'aerodinamica pone le sue basi nello studio dell'idro-dinamica sviluppandosi poi autonomamente e mostrando da subito una elevata complessità dovuta inizialmente alla quasi totale impossibilità di un esame visivo dell'andamento delle correnti d'aria sulla superficie del corpo su cui agiscono le forze aerodinamiche. Il termine fu coniato da P.S. Langley nel 1891 per indicare la parte della meccanica che studia i fenomeni riguardanti il moto dell'aria e quello di corpi in essa immersi, ma in effetti riguardante tali fenomeni in un aeriforme qualunque; in termini generali, è la parte della meccanica dei fluidi relativa agli aeriformi [24]. Alla fine del XIX secolo la letteratura scientifica su questo argomento era ridottissima: si potrebbe affermare che tale importantissimo ramo della scienza non esisteva ancora quando nel dicembre del 1903 il primo aeroplano si levava a volo nell'aria [25]. Fu, in realtà, Newton a dare la prima impostazione razionale dei problemi aerodinamici, ma la complessità della materia renderà i risultati della ricerca di Newton comprensibili solo anni dopo. Negli anni successivi anche D'Alambert, Legendre e Saint-Venant affrontarono il problema, ma i risultati sembravano paradossali per l'epoca. Per questo iniziò ad affermarsi il bisogno delle ricerche sistematiche tecniche e pratiche nel campo dell'aerodinamica. I laboratori sperimentali si moltiplicarono, i matematici crearono una nuova scienza, l'aerodinamica, prelevando metodi e teorie dall'idrodinamica.

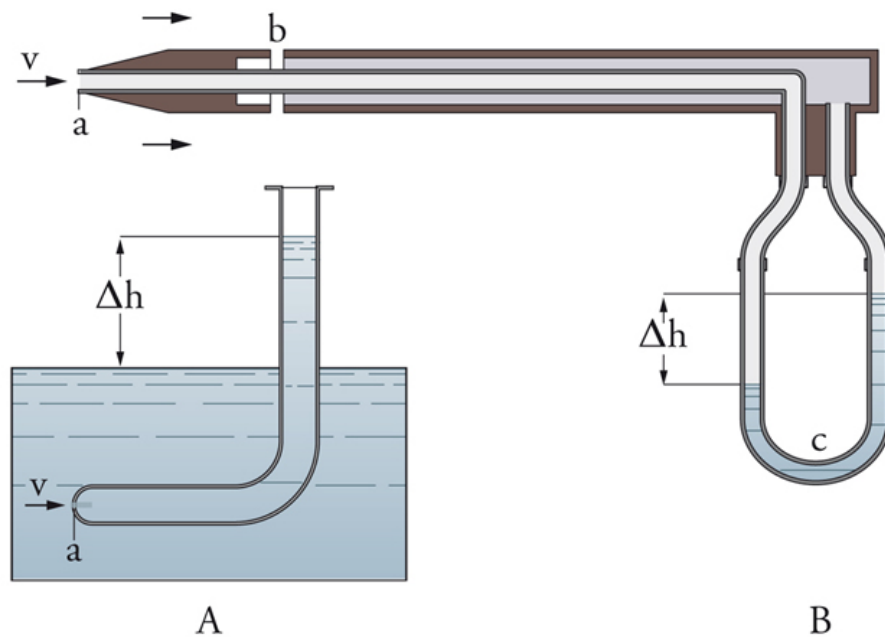


Figura 2.2: Tubo di Pitot

Uno dei primi studi sperimentali risale all'inglese Benjamin Robins che riuscì a inserire la resistenza dell'aria negli studi balistici, migliorando la precisione dei propri cannoni, mentre la prima galleria del vento venne realizzata da Wenham.

Uno dei primi problemi che storicamente venne affrontato fu quello della velocità della corrente d'aria che investe il corpo fermo. Nel 1732 dall'ingegnere francese Pitot fu inventato l'omonimo tubo che consiste in un tubo ripiegato ad L e immerso con la parte ripiegata contro corrente; l'altra parte comunica con un ordinario manometro ad acqua o ad alcool contenuto in un tubo ad U. Questo strumento misura, con il dislivello delle colonne liquide, il valore della pressione locale dovuta al moto del fluido (figura 2.2). Ancora oggi, gli attuali pneumometri utilizzati in galleria del vento, sono derivati dalla tecnologia di questo ingegnere francese.

Per tener conto della pressione statica dell'ambiente in cui si misura la velocità, poiché il manometro è tenuto fuori di questo, sono stati escogitati molti dispositivi: uno di questi consiste nell'avvolgere la parte ripiegata del tubo di Pitot con un cilindro (raccordato all'orifizio del tubo) sulle cui generatrici sono praticate, in opportuna posizione, dei forellini in comunicazione con l'al-

tra estremità del manometro ad U. Viene così automaticamente eliminata la pressione statica che regna nella vena ed il dislivello tra le due colonne segna effettivamente il valore della pressione corrispondente alla velocità.

Nel 1835 i capitani Piobert, Morin e Dedion, attraverso una puleggia, un cordone di seta scorrente e delle lamine piane di forma diversa andarono a determinare la legge del moto di discesa e gli elementi necessari al calcolo. Esperienze di questo tipo furono fatte anche negli anni seguenti, in particolare dell'abate Le Dantec nel 1878 nella cappella del Conservatorio di arti e mestieri di Parigi. Nel 1892 Cailletet e Colardeau eseguirono esperienze dal secondo piano della torre Eiffel nel laboratorio da essi ivi organizzato. Lo stesso Eiffel nel 1903 condusse una serie sistematica di prove su corpi dalle forme più varie, servendosi di un apparecchio più grande e più perfezionato di quello adoperato dai suoi predecessori. L'apparecchio di misura che seguiva il corpo nella caduta, era guidato da un cavo verticale e registrava la resistenza mediante opportune molle antagoniste, ed i tempi mediante un diapason.

Le prime esperienze su locomotive e carrelli furono eseguite in acqua nel 1840 nel canale di S.Mauro da Duchemin mediante una fune tesa a cui erano assicurate le lamine da sperimentare e gli strumenti di misura, e lungo questa fune si faceva muovere il complesso così formato.

Ricour (1885) volle studiare l'effetto della resistenza dell'aria sul rendimento delle locomotive variandone la superficie esterna e iniziò le sue esperienze su lamine perpendicolari alla direzione del moto, poste lateralmente alla locomotiva e tenute da parallelogrammi snodati, adatti alla misura delle forze. Desdouits riprese queste esperienze con tavolette collocate lateralmente al treno, lanciato a velocità note e crescenti.

Anche la società Siemens e Halske nel 1901 fece esperienze con lo stesso scopo e cominciò col determinare la distribuzione delle pressioni su una vettura elettrica, lanciata a 160 km-ora sulla linea ferroviaria tra Zossen e Berlino, mediante tubi sboccanti dalla superficie esterna della vettura e facenti capo a manometri.

Nel 1903 il tenente Arturo Crocco, utilizzando il pneumometro Krell, riuscì a misurare la direzione della velocità. Questo strumento è costituito da un cilindretto cavo separato in due camere, ciascuna delle quali era in comunicazione, attraverso un forellino praticato nel centro, con l'aria della corrente e con i rispettivi tubi partenti dalle due camere e terminanti alle due estremità del manometro. Il tenente osservò che quando le facce del disco giacevano nel letto della corrente, le pressioni ai due fori si equivalevano e il manometro non accusava dislivello, mentre questo diveniva sensibilissimo per una variazione di angolo piccolissima: e costruì così un pneugoniometro, adatto sia a dare la misura della direzione della velocità come intersezione di due giaciture, sia quella della intensità con una rotazione di 90 gradi rispetto alla detta direzione.

Dall'inizio del 1900 si susseguirono diverse esperienze con automobili su pista raggiungendo nel 1907 la velocità di 127 km/h. È in questi anni che il milionario francese Henry Deutsch de la Meurthe, fondò a Saint Cyr presso Parigi, l'Istituto aerotecnico diretto da Maurain e Toussaint (1910), che comprendeva un tunnel a ventilatore, varie officine e una pista rettilinea a rotaie lunga originariamente 1400 m sulla quale scorreva un carrello elettrico capace di raggiungere la velocità di 90 km/h e destinato alle prove al vero di ali, eliche, e di altri corpi. Con questo carrello furono sperimentate eliche ed ali e fu determinata col metodo manometrico la distribuzione topografica delle pressioni. In linea di massima queste esperienze dettero risultati abbastanza concordanti con quelli delle esperienze eseguite in scala più piccola al tunnel Eiffel.

È proprio a questi anni che risalgono i primi "tunnel" che vanno a riprodurre la corrente d'aria necessaria. Il primo in Italia venne costruito nel 1903 da Crocco 2.3 che utilizzò un gassometro ad ossigeno per verificare la sua legge del rettangolo delle velocità. Negli stessi anni ne furono costruite alcune in Inghilterra ed una più grande a Roma. A partire da questi primi esempi la galleria del vento divenne l'elemento fondamentale per gli studi aerodinamici.

Nel 1904 Prandtl rende nota la sua teoria sullo strato limite che permette di valutare gli effetti della viscosità dell'aria confinandoli in un sottile strato che

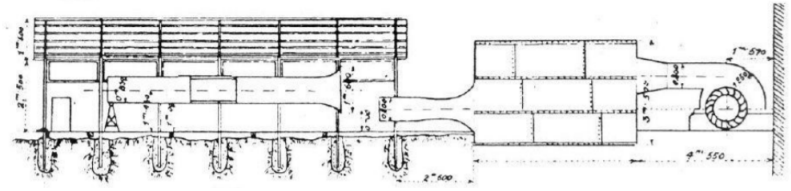


Figura 2.3: La prima galleria di Crocco

aderisce alla superficie del corpo investito dalla corrente.

Nel 1905 Zukovskij pubblica la fondamentale memoria sul fenomeno del vortice portante che giustifica per via teorica la possibilità della sustentazione aerodinamica. Dall'omonimo teorema deriva la definizione dei profili alari attraverso al tecnica delle trasformazioni conformi, che permette di affrontare quest'importantissima parte dell'aerodinamica in maniera rigorosa e sistematica.

Nel 1919 sul campo sperimentale di Montecelio, si predispose un aeroplano per le prove manometriche praticando piccoli fori sul dorso delle ali, sul ventre di queste e sulla superficie esterna della fusoliera; ciascuno di detti fori faceva capo ad una capsula manometrica che registrava su un cilindro affumicato le pressioni verificatesi nella regione nella quale sboccava il foro. Era così possibile, durante il volo e durante le evoluzioni, avere il quadro della distribuzione in ogni istante di queste pressioni e quindi degli sforzi elementari cui fusoliera e velatura erano sottoposte

A metà del XIX secolo Mach definì le basi matematiche dell'aerodinamica delle alte velocità. L'aerodinamica teorica, caratterizzata da sviluppi analitici di estrema complessità, rimase a lungo utilizzabile quasi esclusivamente per lo studio di pochi problemi relativamente semplici, e spesso di scarso interesse applicativo, al punto che fino a non molti decenni or sono il progresso dell'aerodinamica fu essenzialmente dovuto al sistematico ricorso alla sperimentazione nella galleria del vento.

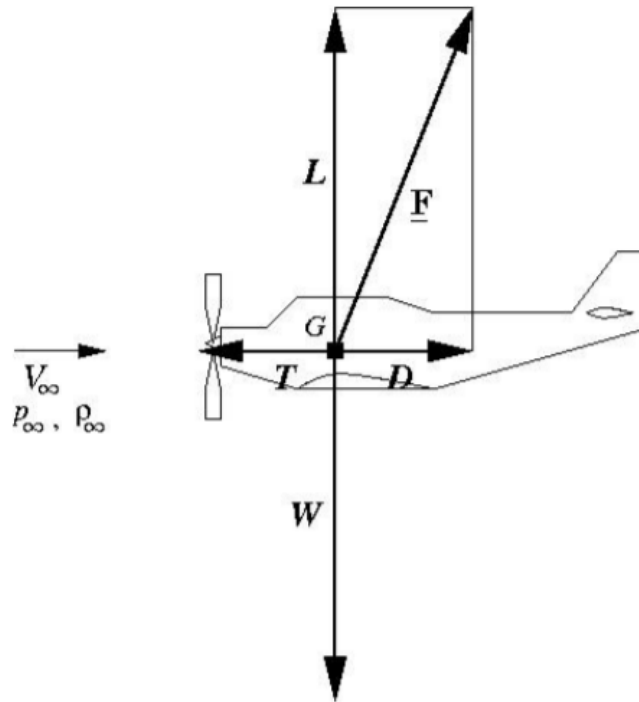


Figura 2.4: Forze che agiscono sul velivolo

2.1.2 Principi fondamentali di aerodinamica

Per andare ad indagare le forze che interagiscono con il veicolo si è soliti far riferimento alla normalizzazione delle grandezze fisiche, questo poichè è necessaria una opportuna correlazione in fase di avamprogetto o durante la procedura trial and error, in sede di simulazione aerodinamica, del comportamento delle superfici, nella fattispecie in galleria del vento.

Per andare a definire le forze aerodinamiche si sceglie un sistema inerziale di riferimento $O(x,y,z)$ solidale con il veicolo che è investito da una corrente uniforme di velocità V_∞ , alla quota h , caratterizzata dalla pressione p_∞ e dalla densità ρ (figura 2.4).

F , le cui componenti sono L e D , rappresenta la *forza aerodinamica*; L è la *portanza* (lift) perpendicolare alla velocità della corrente uniforme e D la *resistenza* (drag) parallela a V_∞ . W rappresenta il peso applicato nel baricentro

G e T la spinta. La forza aerodinamica di riferimento è pari a:

$$F = \frac{1}{2}\rho V_{\infty}^2 S \quad (2.1)$$

Per andare a valutare le prestazioni del veicolo si fa riferimento ai coefficienti di portanza (equazione 2.2) e di resistenza (equazione 2.3):

$$C_l = \frac{L}{\frac{1}{2}\rho V_{\infty}^2 S} \quad (2.2)$$

$$C_d = \frac{D}{\frac{1}{2}\rho V_{\infty}^2 S} \quad (2.3)$$

Dove S è l'area frontale proiettata su un piano della vettura o, nel caso si faccia riferimento alle ali isolate, è la superficie alare in piante; L è la portanza generata, D la resistenza aerodinamica al moto, ρ la densità del flusso che compone la corrente aerodinamica e V_{∞} la sua velocità nella condizione a monte supposta indisturbata.

Un altro elemento fondamentali dell'analisi aerodinamica di un veicolo è il coefficiente di forza laterale Y o side force (equazione 2.4) , che va tenuto in considerazione in caso di forte vento laterale o durante le situazioni di sorpasso.

$$C_y = \frac{Y}{\frac{1}{2}\rho V_{\infty}^2 S} \quad (2.4)$$

Questi tre coefficienti possono essere definiti come C_z , C_x e C_y secondo l'orientamento classico della terna cartesiana sul veicolo. I progetti delle auto da corsa si riferiscono a propri coefficienti rispetto a delle specifiche aree di riferimento e la valutazione dei coefficienti può variare in base ai metodi di misurazione utilizzati. Tuttavia, secondo quanto riportato da Katz [26], per macchine da corsa prototipo possiamo stimare dei valori massimi di $C_l = -3.00/-4.00$ e del $C_d = 0.75$, valore che può anche essere superiore nelle open-wheel race cars per via della resistenza generata dalle ruote scoperte, il contributo può variare da 0.2 a 0.5.

I parametri fondamentali che ci permettono di classificare la corrente sono i seguenti:

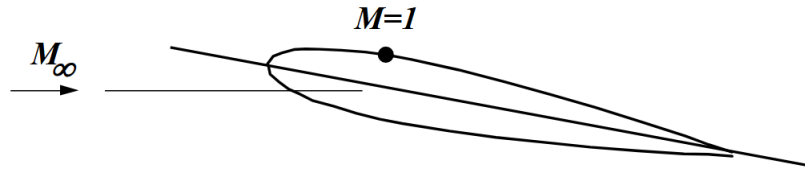


Figura 2.5: Mach critico di un profilo

- **Numero di Mach:** è un parametro utile per definire lo stato dinamico di un fluido. È definito come il rapporto tra la velocità locale del fluido V ed il valore a della velocità locale del suolo:

$$M = \frac{V}{a} \quad (2.5)$$

Rappresenta gli effetti di compressibilità del fluido. Si distinguono le seguenti condizioni:

- $Ma < 0.3$ regime subsonico incomprimibile;
- $0.3 < Ma < 0.7$ flusso con dipendenza non trascurabile dagli effetti di compressibilità, regime subsonico;
- $0.7 < Ma < 1.3$ flusso in regime transonico con dipendenza importante dagli effetti di compressibilità, regime transonico;
- $Ma = 1$ regime sonico;
- $Ma > 1.3$ flusso in regime supersonico con dipendenza fondamentale dagli effetti della compressibilità e con onde d'urto;
- $Ma > 5$ regime ipersonico.

Quando il numero di Mach si avvicina ad 1 allora in qualche punto del profilo saranno raggiunte le condizioni soniche (la velocità sul dorso del profilo è maggiore della velocità a monte), il numero di Mach della corrente M_∞ per il quale si raggiungono le condizioni "soniche" sul profilo per la prima volta viene detto Mach critico (figura 2.5).

- **Numero di Reynolds:** è un parametro adimensionale proporzionale al rapporto tra le forze di inerzia e quelle viscosi:

$$Re = \frac{\rho V l}{\mu} \quad (2.6)$$

con μ viene indicata la viscosità dinamica, con ρ la densità del fluido, con V la velocità di riferimento e con l la lunghezza del fenomeno che si vuole studiare. Questo parametro consente di andare a valutare il regime del flusso se laminare o turbolento e è dipendente dalla geometria dell'oggetto inserito nel fluido.

2.1.3 Aerodinamica e CFD

Per risolvere i complessi problemi fluidodinamici, a partire dagli anni '50, si è sviluppata la fluidodinamica computazionale CFD. Questa tecnica, grazie all'utilizzo di processori che trasformano il problema fisico in problema matematico, permette di ridurre il numero dei test sperimentali condotti su i modelli reali in galleria del vento. I vantaggi di questa tecnica consentono risparmi dal punto di vista temporale ed economico.

La CFD si compone di 3 fasi:

1. **processore:** il problema fisico viene "tradotto" in un problema matematico. Viene definita una mesh che permette di andare a discretizzare il dominio di calcolo della particolare geometria di interesse. Vengono poi definite le condizioni al contorno per il problema e le proprietà del fluido;
2. **solutore:** provvede all'integrazione numerica delle equazioni del moto del fluido, risolvendo iterativamente il sistema algebrico. Il metodo utilizzato può essere quello dei volumi finiti, degli elementi finiti o delle differenze finite.
3. **postprocessore:** vengono analizzati ed elaborati i risultati del passaggio precedente.

L'analisi fluidodinamica dipende dalle equazioni del moto del fluido e dalle caratteristiche dello stesso; in particolare, grazie ai parametri visti nel paragrafo 2.1.2, possiamo andare a definire i regimi fluidodinamici. Quello più semplice è il regime laminare, che si realizza a bassi numeri di Reynolds. In esso prevalgono gli effetti viscosi, i quali inibiscono i trasferimenti di energia tra le strutture più grandi del sistema e quelle più piccole, impedendo la formazione di effetti di turbolenza. A questo si contrappone il regime di moto turbolento, nel quale invece prevalgono gli effetti d'inerzia, per cui si hanno i succitati trasferimenti di energia tra strutture grandi e piccole. Vi è poi infine la situazione intermedia, alla quale ci si riferisce come "zona di transizione". La maggior parte dei problemi di fluidodinamica coinvolge fenomeni di natura turbolenta, i quali si caratterizzano per l'intrinseca non linearità, per la loro natura caotica e per la forte dipendenza dalle condizioni al contorno.

Una volta definito il regime, per qualsiasi tipo di flusso il solutore risolve le equazioni della conservazione della massa, della quantità di moto e dell'energia oltre all'equazione di stato (equazioni di Navier-Stokes):

$$\left\{ \begin{array}{l} \frac{\partial p}{\partial t} + \nabla \cdot (\rho u) = 0 \\ \frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) = -\nabla p + \nabla \cdot \tau \\ \frac{\partial(\rho e_t)}{\partial t} + \nabla \cdot (\rho e_t u) = -\nabla \cdot q - \nabla \cdot (\rho u) + \tau : \nabla u \\ p = \rho R_g T \end{array} \right. \quad (2.7)$$

dove τ è il tensore relativo alla scia.

Qualora ci si trovasse in condizione di flusso turbolento le variazioni di quantità di moto sono governate dalle forze viscosi e dagli scambi di energia tra il flusso principale e i vortici, nonché tra gli stessi vortici più grandi e più piccoli secondo il modello "a cascata". I flussi in condizioni turbolente sono flussi ro-

tazionali con $\omega = \nabla \times v \neq 0$, essi sono di tipo tridimensionale non stazionario. Poiché in questo caso le equazioni di Navier-Stokes sono altamente non lineari, variazioni minime delle condizioni al contorno comportano elevate variazioni di velocità e pressione; per aggirare questo problema si dovrebbe infittire la griglia computazionale col fine di riuscire a simulare anche l'andamento dei vortici più piccoli. Una soluzione analitica dei flussi turbolenti si riesce ad avere solo nel caso di flussi molto semplici, per cui è necessario usare dei modelli di turbolenza che ne prevedono il comportamento per riuscire ad avere una soluzione delle equazioni di Navier-Stokes.

Se si indica con $a(x, t)$ una grandezza caratteristica di un flusso turbolento funzione della posizione e del tempo, essa può essere rappresentata come somma di più contributi:

- grandezza media $\bar{a}(x, t)$;
- grandezza fluttuante $a'(x, t)$

e ipotizzando la densità costante, si può, quindi, esprimere le velocità e la pressione come segue:

$$u(x, t) = \bar{u}(x, t) + u'(x, t) \quad (2.8)$$

$$p(x, t) = \bar{p}(x, t) + p'(x, t) \quad (2.9)$$

In questo caso le equazioni di Navier-Stokes vengono riscritte nella forma nella quale le grandezze fluidodinamiche sono mediate nel tempo e non più istantanee e sono note come RANS (*Reynolds Averaged Navier Stokes*).

$$\begin{cases} \frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla) \bar{u} = -\frac{1}{\rho} \nabla \bar{p} + \nu \nabla^2 \bar{u} - \nabla \cdot (\overline{u'u'}) \\ \nabla \cdot \bar{u} = 0 \end{cases} \quad (2.10)$$

con $\nu = \mu/\rho$

Queste equazioni vettoriali costituiscono un sistema di 4 equazioni scalari in dieci incognite: esse sono le 3 componenti della velocità, la pressione media e le sei componenti fluttuanti

$$\overline{u'u'} = \begin{bmatrix} \overline{u_x'^2} & \overline{u_x'u_y'} & \overline{u_x'u_z'} \\ \overline{u_x'u_y'} & \overline{u_y'^2} & \overline{u_y'u_z'} \\ \overline{u_x'u_z'} & \overline{u_y'u_z'} & \overline{u_z'^2} \end{bmatrix} \quad (2.11)$$

Queste equazioni devono essere scritte in funzione di altre incognite. Considerando che i flussi turbolenti aumentano la dissipazione di energia nel tempo di un fluido, i termini fluttuanti possono essere pensati come degli sforzi viscosi aggiuntivi agenti sul fluido. Si definisce, quindi, tensore degli sforzi di Reynolds:

$$\overline{\rho u'u'} = \overline{\overline{\tau}} \quad (2.12)$$

Sostituendo la 2.12 alla 2.6 si ottiene:

$$\begin{cases} \frac{\partial \overline{u}}{\partial t} + (\overline{u} \cdot \nabla) \overline{u} = -\frac{1}{\rho} \nabla \overline{p} + \nu \nabla^2 \overline{u} - \frac{1}{\rho} \nabla \cdot \overline{\overline{\tau}} \\ \nabla \cdot \overline{u} = 0 \end{cases} \quad (2.13)$$

Il tensore $\overline{\overline{\tau}}$ può essere rappresentato tramite modelli di tipo algebrico ARSM (*Algebraic Reynolds Stress Model*) o su modelli differenziali RSTM (*Reynolds Stress Transport Model*).

Un classico modello algebrico lineare è quello di *Bussinesq*, costituito in analogia alla *legge di Stokes*, che rappresenta il tensore degli sforzi viscosi.

Questo modello porta ad un sistema di dieci equazioni scalari in dodici incognite:

$$\begin{cases} \frac{\partial \overline{u}}{\partial t} + (\overline{u} \cdot \nabla) \overline{u} = -\frac{1}{\rho} \nabla \overline{p} + \nu \nabla^2 \overline{u} - \frac{1}{\rho} \nabla \cdot \overline{\overline{\tau}} \\ \overline{\overline{\tau}} = \frac{2}{3} \rho k [I] - 2 \mu_t \overline{\overline{S}} \\ \nabla \cdot \overline{u} = 0 \end{cases} \quad (2.14)$$

con $[I]$ è indicata la matrice identità, con $\overline{\overline{S}}$ il flusso medio, ovvero il tensore velocità di deformazione, e con μ_t la viscosità turbolenta. Questa ultima variabile rappresenta una delle ulteriori incognite del problema insieme all'energia

cinetica turbolenta per unità di massa:

$$k = \frac{1}{2} \overline{u'u'} \quad (2.15)$$

Il 95% di tale energia cinetica turbolenta è presente nei vortici più grandi, pertanto è possibile far riferimento unicamente a questi ultimi per descrivere k e μ_t , dopo averne definito una opportuna scala per le velocità indicata con v^* ed una scala per i tempi indicata con t^* . Avendo due scale il modello è classificato come modello a due equazioni. I modelli di turbolenza più comuni prevedono la risoluzione di una equazione per k ; fra i vari modelli cambia la metodologia con cui viene indicata la scala dei tempi t^* . Di seguito un elenco di alcuni modelli di turbolenza:

- **modello di turbolenza Spalart-Allamaras:** è un semplice modello ad una equazione, il quale procede alla risoluzione della sola equazione di trasporto per la viscosità (*eddy viscosity* [27]), la quale differisce dalla viscosità cinematica turbolenta unicamente in prossimità della parete. Il modello è del tipo eddy-viscosity nel senso che si ritiene valida l'ipotesi di Bussinesq, ed è completo in quanto definisce automaticamente la scala di lunghezza della turbolenza [28]. Il modello è stato sviluppato specificamente per applicazioni aerospaziali, e ha fornito prova di buone capacità nella risoluzione di flussi caratterizzati da *Boundary-Layers* soggetti a gradienti avversi di pressioni e in diverse applicazioni sulle turbomacchine.
- **modello di turbolenza $k - \varepsilon$:** costituisce uno dei più classici modelli a due equazioni, nel quale cioè si procede alla risoluzione di due separate equazioni di trasporto [29]. Si tratta di un modello abbastanza robusto e preciso, ma di accuratezza limitata nella risoluzione dei flussi vicino a parete e in presenza di forti gradienti di pressione.
- **modello di turbolenza $k - \omega$:** è simile al modello precedente. La frequenza caratteristica dei vortici, o dissipazione turbolenta specifica, ω viene definita come il rapporto fra la velocità di dissipazione ε e l'energia

cinetica turbolenta. Le equazioni di trasporto sono caratterizzate dalla presenza di un termine dissipativo, uno di diffusione ed uno di produzione. Questo metodo presenta il vantaggio di una più semplice integrazione in quanto non presenta particolari problematiche in prossimità della parete ed è in grado di lavorare con deboli gradienti di pressione. Presenta, però, una forte sensibilità ai valori caratteristici del flusso libero.

Le equazioni oggetto di studio sono poi risolte numericamente tramite la griglia di calcolo che permette di discretizzare il dominio di riferimento.

2.2 L'aerodinamica e la F1

Negli anni sessanta nel mondo delle corse si è compreso che le prestazioni potevano essere migliorate attraverso la gestione aerodinamica. Da quel momento in poi il settore ha avuto un fortissimo sviluppo con l'obiettivo di andare sempre più a migliorare le prestazioni del veicolo.

L'importanza dell'aerodinamica in F1 fu palese nel famosissimo incidente del 1969 quando una rottura dell'ala fece alzare la vettura del pilota Graz che descrive così la vicenda: *"...quando l'alettone posteriore si ruppe e le ruote dietro si sollevarono da terra, la macchina sterzò bruscamente in direzione del guardrail alla mia sinistra, rimbalzò verso quello opposto, andò a sbattere contro i resti della vettura di Graham e si ribaltò. Fortunatamente i commissari girarono l'auto e furono veloci ad estrarmi, ma il mio viso era visibilmente pieno di tagli ed inoltre mi ero procurato anche una microfrattura del cranio"*. È a valle di questo incidente che il "popolo della Formula 1" capì quanto carico poteva essere generato da una vettura da corsa e come questo potesse incidere sulle prestazioni, sulla stabilità della monoposto e anche sulla sicurezza del pilota [30]. Una delle innovazioni introdotte recentemente nella Formula 1 è il DRS ¹. Come suggerisce anche il nome, è un dispositivo aerodinamico con il compito di ridurre la resistenza aerodinamica della vettura e quindi aumentare la ve-

¹DRS: Drag Reduction System, sistema di riduzione della resistenza

locità massima di circa 5-10km/h. Si tratta di un meccanismo di rotazione che riduce l'incidenza del profilo principale dell'ala posteriore, può essere attivato dal pilota tramite un comando sul volante e si chiude automaticamente in frenata. L'utilizzo di questo sistema è stato reintrodotta solo nel 2011 a seguito di numerosi test ed oggi è utilizzabile solo in alcuni tratti del circuito per facilitare i sorpassi.

Vista l'importanza dell'aerodinamica ogni team di formula 1 ha un suo dipartimento specifico che si occupa nei minimi dettagli, sempre nel rispetto del regolamento, di tutte le superfici che entrano in contatto con l'aria. Gli aerodinamici però non si occupano solo di migliorare le prestazioni dell'auto, ma devono anche badare a come incanalare l'aria per raffreddare alcune componenti come il motore, i freni, i radiatori, l'elettronica. L'aerodinamica delle auto viene aggiornata non solo di anno in anno, cercando di raggiungere sempre prestazioni migliori, ma è adeguata alle diverse caratteristiche dei circuiti e alle variazioni delle condizioni meteo in ogni gara.

I progressi aerodinamici sono stati molteplici e le moderne auto di Formula 1 hanno tanto in comune sia con un aereo quanto con una normale auto da strada. L'aerodinamica è diventata un punto cardine ed influenza diversi aspetti delle monoposto, dalla progettazione di una componente aerodinamica fino alla gestione dei flussi per raffreddare le parti meccaniche. È infatti una chiave per il successo nello sport e le squadre spendono ogni anno milioni di dollari in ricerca e sviluppo sul campo per migliorare le prestazioni. Il progettista aerodinamico ha due preoccupazioni principali [31]:

1. la creazione di carico aerodinamico, per aiutare a spingere gli pneumatici dell'auto sulla pista e migliorare le forze durante la percorrenza in curva,
2. la riduzione al minimo della resistenza all'avanzamento della monoposto.

La filosofia di base è quindi quella di massimizzare il carico aerodinamico e minimizzare la resistenza prodotta dalla vettura mantenendo un bilancio aerodinamico ragionevole lungo tutto l'intervallo di funzionamento della macchina.

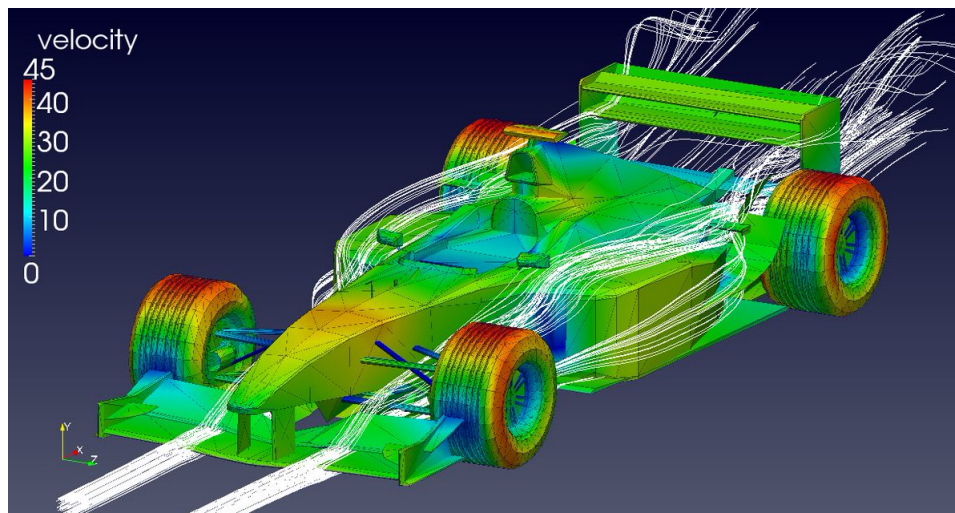


Figura 2.6: Auto con forze aerodinamiche indirette

Per quanto sia un'idea semplice, è complesso ottenerla nella realtà [32]. Il risultato finale è quindi un compromesso tra i diversi fattori contrastanti. A causa della natura stessa delle auto di F1 i coefficienti aerodinamici sono molto diversi rispetto a quelli di una auto stradale, questo in parte è dovuto alle ruote esposte e in parte ai benefici della deportanza. Le ali che generano deportanza cercano di sfruttare anche il *ground effect* poichè quelle vicine al suolo sono più efficienti dato che producono meno resistenza a parità di carico aerodinamico di quelle molto lontane dal suolo. Per quanto riguarda l'ala anteriore il design ottimale è dettato dall'influenza che l'ala ha sul flusso verso il resto dell'auto piuttosto che dalla necessità diretta di deportanza anteriore [30]. Non sempre i dispositivi che i team utilizzano apportano contributi diretti al carico aerodinamico, alcuni elementi hanno il compito di gestirne i flussi in maniera tale da provocare delle forze aerodinamiche indirette che migliorano però le prestazioni complessive della vettura. Nella figura 2.6 si può vedere come alcune linee di flusso vengono deviate dalle varie appendici non solo per generare deportanza ma anche per indirizzare i flussi in zone dove altre appendici vanno a generare carico aerodinamico [30].

Nella figura 2.7 si può notare l'andamento del carico aerodinamico e della resistenza in una vettura prima del 2009 (quando le regole erano diverse da

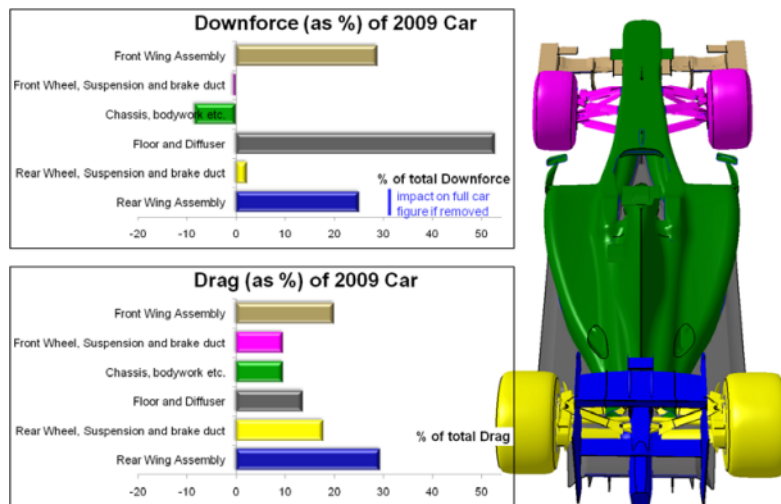


Figura 2.7: Carico aerodinamico e resistenza delle componenti principali nel 2009

quelle attuali)[30].

Per quanto riguarda la resistenza e la sua natura per le monoposto di Formula 1, questa differisce dal caso aeronautico in cui metà di questa è dovuta agli effetti viscosi di strato limite e metà è indotta dalla portanza: infatti il contributo della resistenza di forma e indotta è pari al 90% della totale [30].

Nel mondo della formula uno si è sviluppato un utilizzo massiccio della CFD, diventando un punto di riferimento per l'aerodinamica delle auto da corsa. Le squadre hanno approfondito l'uso della CFD alla fine degli anni '80 e all'inizio degli anni '90 quando era già ampiamente utilizzata nell'industria aeronautica.

Individuate le singole componenti dell'auto si impostano la mesh e le simulazioni. La CFD ci permette di andare a valutare i contributi di ogni singolo elemento dell'autovettura e di andare ad analizzare le modifiche da fare o effettuate. Possono essere riassunti i contributi di deportanza di ogni singola componente del veicolo come riportati ad esempio nella tabella 2.1 (con SC_z viene indicata la componente del carico verticale)[33]. Il fondo dell'auto, composto dal sottoscocca e dalla plancia, genera più o meno il 58% del carico aerodinamico complessivo, mentre l'ala anteriore e quella posteriore forniscono rispettivamente il 26,3% e il 27,5% del contributo totale (tabella 2.1).

Il contributo alla resistenza (SC_x) dei principali componenti dell'auto è vi-

Component	SC_z [m^2]	Contribution [%]
Cockpit	+0.04	+1.2
Driver	+0.02	+0.6
Front bodywork	-0.10	-3.0
Front suspension	+0.06	+1.8
Front wing	-0.88	-26.3
Plank	-0.45	-13.4
Rear suspension	-0.01	-0.3
Rear wing	-0.92	-27.5
Sidepod	+0.2	+6.0
Underbody	-1.49	-44.5
Upper bodywork	+0.18	+5.4
Full car	-3.35	≈ 100

Tabella 2.1: Contributo di deportanza delle singole componenti del veicolo

sibile nella tabella 2.2, si può vedere come le ruote siano responsabili del circa 30% della resistenza totale [33].

Component	SC_z [m^2]	Contribution [%]
Cockpit	+0.04	+1.2
Driver	+0.02	+0.6
Front bodywork	-0.10	-3.0
Front suspension	+0.06	+1.8
Front wing	-0.88	-26.3
Plank	-0.45	-13.4
Rear suspension	-0.01	-0.3
Rear wing	-0.92	-27.5
Sidepod	+0.2	+6.0
Underbody	-1.49	-44.5
Upper bodywork	+0.18	+5.4
Full car	-3.35	≈ 100

Tabella 2.2: Contributo della resistenza delle singole componenti del veicolo

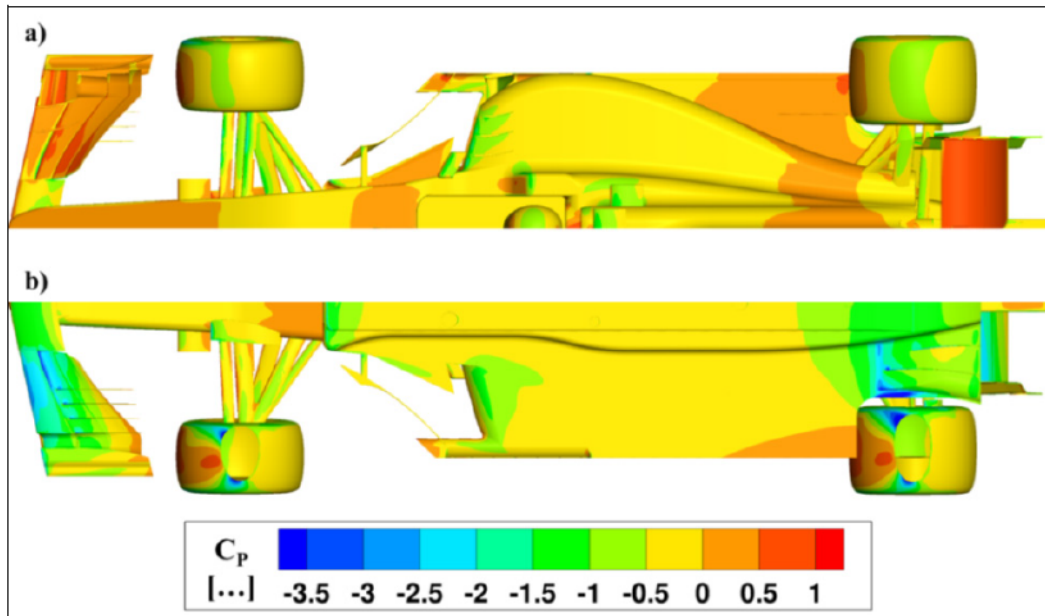


Figura 2.8: Coefficiente di pressione: a vista dall'alto, b vista dal basso

La stessa analisi CFD ci permette di andare ad analizzare e illustrare il coefficiente di pressione (C_p), in particolare nella foto 2.8 [33] possiamo vederlo rappresentato sulla superficie dell'auto. Il fondo della carrozzeria è caratterizzato dalle tipiche "low-pressure cores" che si trovano all'inizio dell'asse, dove l'altezza da terra è minima, all'ingresso del sottoscocca e del diffusore posteriore. In prossimità del disturbo dello pneumatico posteriore, la pressione aumenta e si perdono i benefici dell'effetto suolo. La vista dall'alto mostra il contributo alla deportanza della carrozzeria anteriore, dovuto alla forma del muso e alla zona di ristagno davanti all'abitacolo. Per quanto riguarda le ali, si può notare che l'ala posteriore genera deportanza principalmente per l'elevato angolo del profilo alare; al contrario l'ala anteriore sfrutta l'effetto suolo per accelerare il flusso. Sia la generazione della deportanza che della resistenza indotta sono strettamente connesse con la gestione della velocità assiale [33]. Se prendiamo una rappresentazione grafica della figura 2.9 si può notare che le zone ad alta pressione si trovano nel naso (specialmente nell'ala anteriore) e nell'ala posteriore grazie alla sua elevata curvatura. Alcune zone di ristagno si trovano intorno al cockpit, dove la distribuzione della pressione è bassa e regolare [34].

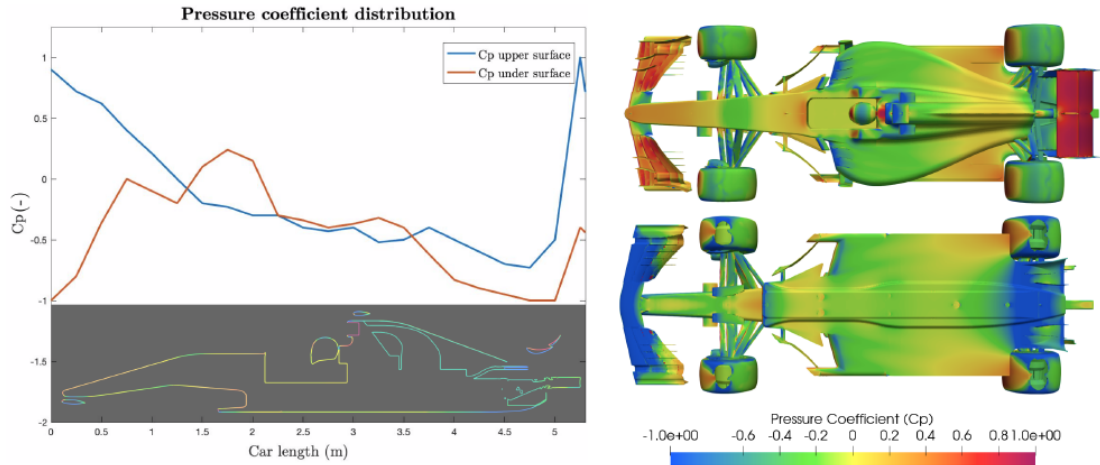


Figura 2.9: andamento del coefficiente di pressione

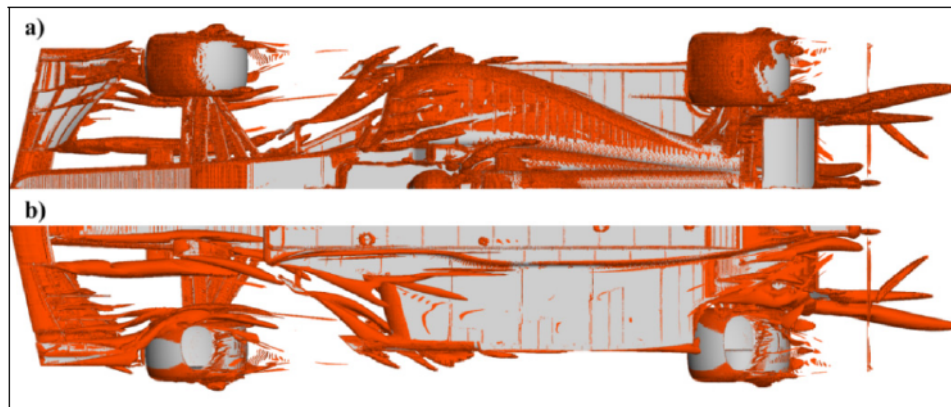


Figura 2.10: Iso-contour con $Q = 50.000l/s^2$. a) vista dall'alto, b) vista dal basso

Attraverso l'analisi fluidodinamica computazionale è possibile studiare anche i singoli vortici che vengono generati attorno all'auto come si può vedere nella figura 2.10 e 2.11 pubblicate nello studio di Ravelli e Savini [33].

La rappresentazione tridimensionale delle vorticosità riportata in figura 2.12 [34] permette di vedere come il flusso si comporta attorno alle diverse aree della vettura e identificare se il flusso risulta caotico o turbolento [34].

L'analisi CFD, quindi, ci permette di andare a valutare i parametri visti nel paragrafo 2.1.2, in particolare la portanza, la resistenza, l'efficienza aerodinamica e il bilanciamento anteriore. Come si è visto la mole di dati è elevata poichè si vanno ad analizzare tutte le singole componenti in numerose configurazioni. Per questo motivo affiancare il *machine learning* alle attività degli aerodinamici

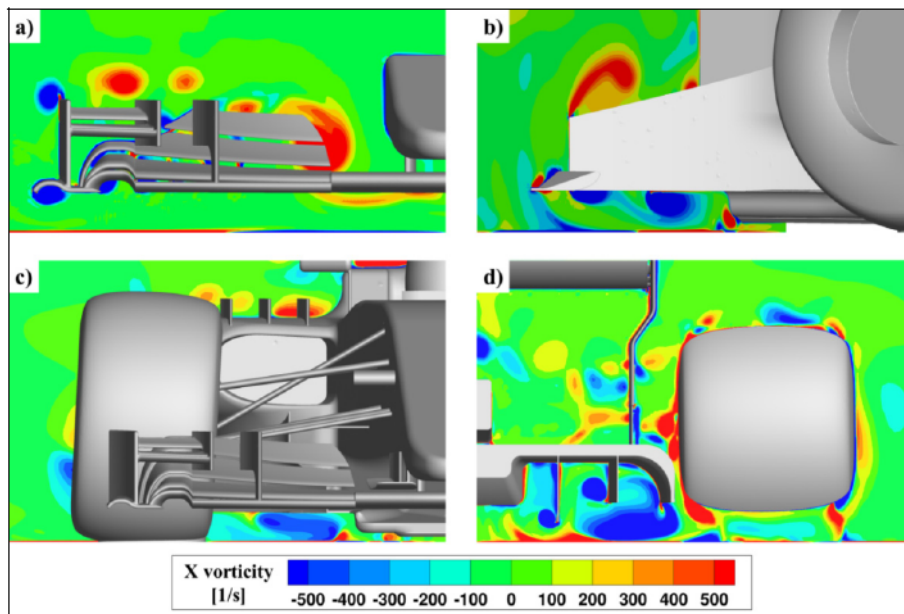


Figura 2.11: vortici assiali. a) ala frontale, b) bargeboard, c) sotto l'ala anteriore, d) diffusore

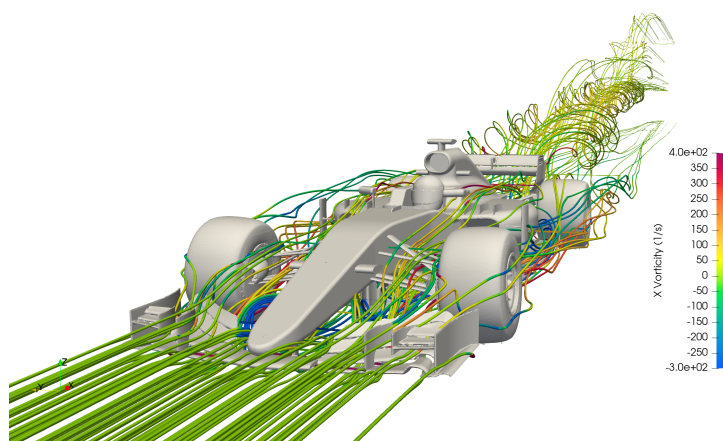


Figura 2.12: Andamento della vorticità assiale

renderebbe il lavoro più veloce e meglio organizzato.

2.3 Il machine learning a supporto degli aerodinamici

Come già detto nei paragrafi precedenti il *machine learning* crea degli automatismi che consentono di velocizzare il lavoro dell'utente umano. Lo sviluppo di questa tecnica di apprendimento meccanico è stata molto utilizzata in altri ambiti e negli ultimi anni si è diffusa anche nel mondo della Formula 1.

Durante il lavoro degli aerodinamici la mole di dati da analizzare, sia proveniente dalla pista che dalla galleria del vento, è cospicua, per questo motivo essere affiancati da software di intelligenza artificiale che siano in grado di imparare ed aiutare, non solo a categorizzare i dati, ma anche ad elaborarli, è indispensabile. Durante la gara elaborazioni sullo stato delle varie componenti, sulla pressione degli pneumatici, sul surriscaldamento dei freni consentono di optare per strategie più o meno vincenti.

Oltre ad attività meramente pratiche durante la gara e di elaborazione in galleria del vento, capita spesso di dover analizzare immagini di squadre concorrenti per studiare le varie linee di sviluppo e confrontarle con quelle della propria scuderia. Questo processo può essere molto lento e dispendioso per gli aerodinamici per cui si è deciso di trovare un modo per fornire loro delle immagini già indicizzate così da poter effettuare la ricerca partendo da parole chiave e restringendo il numero di immagini mostrate all'utente. L'obiettivo del lavoro di tesi è stato quello di velocizzare questo lavoro di scansione e analisi delle immagini da parte degli aerodinamici.

Il programma sviluppato durante questo lavoro permette appunto l'indicizzazione delle foto consentendo un miglioramento nella navigazione e nella ricerca di specifiche componenti aerodinamiche. La disponibilità di queste foto permette un confronto con le altre scuderie e una attenta analisi di come l'andamento delle migliorie tecniche possa andare a modificare le prestazioni della autovettura.

La peculiarità dei software di *machin learning* implica che la programmazione debba essere effettuata da personale in grado di "insegnare" alla macchina tutte le conoscenze aerodinamiche necessarie. È questo il motivo per il quale la programmazione non può essere lasciata a degli informatici ma deve essere effettuata da persone che abbiano conoscenze approfondite di aerodinamica. Lo studio di questo settore dell'ingegneria dovrebbe necessariamente comprendere anche l'informatica, poiché solo una persona competente in entrambi i campi può rendere il *machine learning* uno strumento al servizio della aerodinamica.

Capitolo 3

Il software elaborato

Uno dei compiti dell'aerodinamico è quello di studiare tutte le componenti della macchina, confrontare le performance e adottare soluzioni adeguate anche attraverso il confronto con gli accorgimenti messi in atto da altre scuderie. Questo comporta un lungo lavoro di analisi e ricerca. Gli strumenti a disposizione del dipartimento sono diversi ed uno di questi è l'utilizzo di foto fatte da fotografi professionisti durante gli eventi di Formula 1. I fotografi effettuano un numero elevatissimo di foto in sequenza per poter restituire agli aerodinamici il numero massimo possibile di immagini di alta qualità. Ci si ritrova quindi davanti ad una grande quantità di dati (circa 20GB per ogni sessione di ogni evento) da classificare e conservare.

Solitamente la conservazione delle immagini viene effettuata usando delle cartelle con il nome dell'evento e catalogandole per ogni team, tutto in maniera manuale. Una volta caricate con questa struttura nel server, gli aerodinamici sono liberi di andare a sfogliare le foto ed analizzarne i dettagli. Questa ricerca richiede però moltissimo tempo perché nella cartella di ogni team ci sono tutte le immagini di quell'evento quindi per individuare una componente specifica si deve attendere che si carichi la preview di tutte le foto (dato che le foto sono su un server, questo caricamento può richiedere diversi secondi). Di conseguenza gli aerodinamici passano più tempo nella ricerca delle foto che nell'analisi effettiva di queste.

Valutando queste problematiche si è quindi deciso di trovare una via più

veloce per caricare le immagini e dare agli aerodinamici qualcosa di molto più rapido da utilizzare. L'idea iniziale è stata quella di un programma di categorizzazione delle immagini e per fare ciò potevano esserci diverse soluzioni. Si è deciso di scrivere tutto il codice internamente e usare python come linguaggio di programmazione. Stabilito questo aspetto c'è stato un lavoro di ricerca e documentazione sui programmi già presenti in letteratura. Dopo un'analisi dettagliata si è notato che i più comuni e semplici da utilizzare si basavano sull'utilizzo di database già allenati. Durante la scrittura e lo sviluppo del programma è emerso che i set preallenati non erano funzionali al tipo di catalogazione richiesta dagli aerodinamici perchè dovevano andare a distinguere tra componenti molto specifiche della vettura, mentre questi algoritmi sono utili solo nei casi in cui si stia scrivendo un programma per il riconoscimento di immagini comuni. Si è deciso quindi di creare un set di allenamento ad hoc per il nostro caso e ciò ha richiesto un ingente numero di ore di lavoro con un conseguente aumento esponenziale dei tempi necessari per lo sviluppo del programma. Per raggiungere l'obiettivo prefissato dell'80% di affidabilità è stato necessario categorizzare manualmente oltre 5000 immagini diverse con il corretto componente della vettura che era raffigurato nella foto. Una volta completato il *software* che assegnava automaticamente le etichette alle immagini, ci si è trovati di fronte alla scelta di trovare un programma in grado di restituire all'utente finale una interfaccia grafica dove poter effettuare la ricerca delle foto e visualizzare il risultato.

Le scelte a questo punto erano due, usare qualcosa di interno anche a costo di scrivere un nuovo programma da zero, oppure servirsi di qualche *digital asset management*¹ esterno. Sono state prese in considerazione diverse piattaforme, tra cui Cloudinary [35], Daminion [36] e Razuna [37]. Dopo diversi confronti e suggerimenti si è deciso di affidarsi a Sharepoint [38]. Questa piattaforma veniva già utilizzata dal dipartimento di aerodinamica ed appena si è avuta l'approvazione da parte del dipartimento di sicurezza informatica della Haas si

¹DAM: Digital Asset Management, sono dei software centrati alla gestione dei contenuti, permettono di creare, organizzare e condividere velocemente e facilmente diversi contenuti tra cui le immagini

è iniziato ad utilizzarla per la condivisione, organizzazione e modifica di foto. Sharepoint é un'interfaccia web di Microsoft in grado di fare da database e contemporaneamente permette di effettuare delle ricerche in base a delle etichette, il rallentamento di questa interfaccia è dovuto al tempo di *upload* in quanto bisogna prima inviare un file manualmente con tutte le classi di ogni foto, solo successivamente si può passare a caricare tutte le immagini. Tramite una routine preimpostata la pagina web effettua un controllo per ogni immagine caricata e, se è presente l'etichetta nel file inviato all'inizio, allora inserisce la foto in quella classe. Ci si è resi conto che il caricamento delle immagini ad alta risoluzione nel database online richiedeva tantissimo tempo, si è deciso di caricare delle immagini con una risoluzione inferiore ma con la possibilità di essere reindirizzati, se l'utente finale lo avesse ritenuto opportuno, alla foto presente nel server locale. In questo modo il tempo è stato notevolmente ridotto senza pregiudicare la qualità del lavoro.

3.1 Il modello

3.1.1 Python

Il Python [39] è un linguaggio di programmazione moderno, dalla sintassi semplice e potente che ne facilita l'apprendimento. Gli ambiti di applicazione di questo linguaggio di programmazione sono svariati: sviluppo di siti o applicazioni Web e *desktop*, realizzazione di interfacce grafiche, amministrazione di sistema, calcolo scientifico e numerico, database, giochi, grafica 3D, eccetera. É un linguaggio di programmazione dinamico orientato agli oggetti e utilizzabile per molti tipi di sviluppo *software*. Offre un forte supporto all'integrazione con altri linguaggi e programmi, è fornito di una estesa libreria standard.

Python gira su Windows, Linux/Unix, Mac OS X, OS/2, Amiga, palmari Palm e cellulari Nokia; è stato anche portato sulle macchine virtuali Java e .NET. è distribuito con licenza Open-Source approvata dalla OSI; il suo utilizzo è gratuito e libero anche per prodotti commerciali.

I motivi per cui viene scelto come linguaggio di programmazione sono fondamentalmente 5:

- facilità di memorizzazione e utilizzo: la struttura del linguaggio è di semplice memorizzazione,
- funziona su più piattaforme: può essere utilizzato su tutti i sistemi operativi, uno script ideato per una piattaforma in linea di massima può essere usato anche su altre;
- versatile: ha una infinità di applicazioni;
- librerie incluse: esistono diverse librerie e *framework* che consentono l'utilizzo di questo linguaggio di programmazione in ogni ambito;
- sviluppo: è sicuramente il linguaggio che si sta sviluppando maggiormente, questo non solo permette sempre una maggiore ampiezza delle librerie, ma permette anche il crescere di questa tecnologia in ambito lavorativo.

La libreria più importante che è stato necessario importare per il corretto funzionamento del programma è Tensorflow [8]. Questa è una libreria *software open source* per l'apprendimento automatico, contiene sia moduli ottimizzati e quindi già preallenati che moduli sperimentati, utili nella realizzazione di algoritmi per diversi tipi di compiti percettivi e di comprensione.

3.1.2 Rete neurale convoluzionale

Il cuore di un programma di *machine learning* sono le reti neurali, nel caso in esame sono stati utilizzati 3 livelli di *Convolution Neural Network* (alle quali si farà riferimento da ora in poi per brevità come CNN, figura 3.1). Sono reti neurali specializzate nel processamento di dati che presentano una struttura a griglia, rappresentano un importante passo in avanti nel riconoscimento di immagini, sono la base di ogni modello di *machine learning* e possono essere trovate in qualsiasi algoritmo di classificazione immagini (anche nei software delle auto a guida autonoma), dalla salute fino alla sicurezza. All'interno della

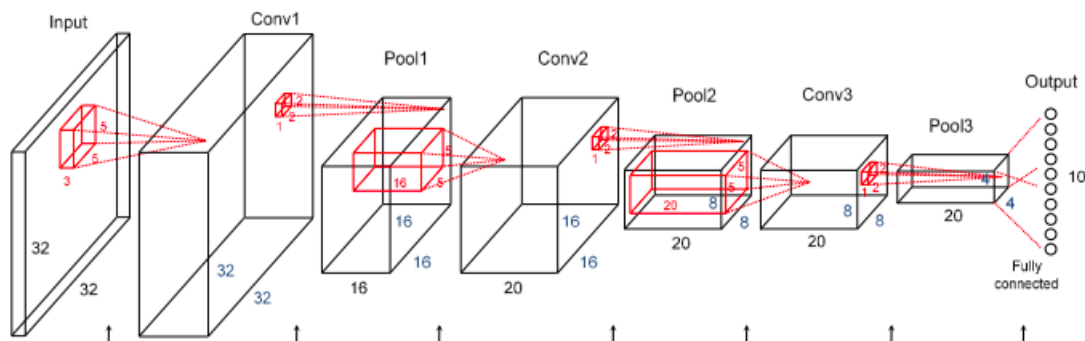


Figura 3.1: Rappresentazione di una possibile rete neurale convoluzionale con 3 livelli di convoluzione e 3 di pooling

rete neurale convoluzionale ci sono diversi livelli con i relativi parametri da impostare, nel dettaglio abbiamo:

- *Livello di input*: è un insieme di numeri che rappresentano, per il computer, l'immagine da analizzare. Essa è rappresentata come un insieme di pixel. Ad esempio nella figura 3.1 il livello di input è una matrice $32 \times 32 \times 3$ dove il primo numero sta ad identificare la larghezza, il secondo l'altezza e il terzo la profondità. Per identificare il colore si usa il formato RGB² ed è per questo che la profondità di una foto colorata è sempre 3, una foto in scala di grigi avrebbe profondità 1.
- *Livello convoluzionale (Conv2D)*: è il livello principale della rete, l'obiettivo è quello di individuare degli "schemi" (come ad esempio figure geometriche, linee, angoli,...) all'interno delle immagini con elevata precisione. Possono essere più di uno ed ognuno si concentra sulla ricerca di queste caratteristiche, maggiore è il loro numero e maggiore è la completezza delle caratteristiche che riescono ad identificare. Ad esempio, si fa scorrere un piccolo filtro (nella figura 3.1 è una matrice $5 \times 5 \times 3$) sulle diverse posizioni dell'immagine di *input*, per ogni posizione viene generato un *output* eseguendo il prodotto scalare tra il filtro e la porzione dell'*input* che viene coperta. In uscita si ottiene un'immagine più piccola che è stata "carat-

²RGB: Red Green Blue, sono i colori rosso, verde e blu usati come base per costruire tutte le tonalità

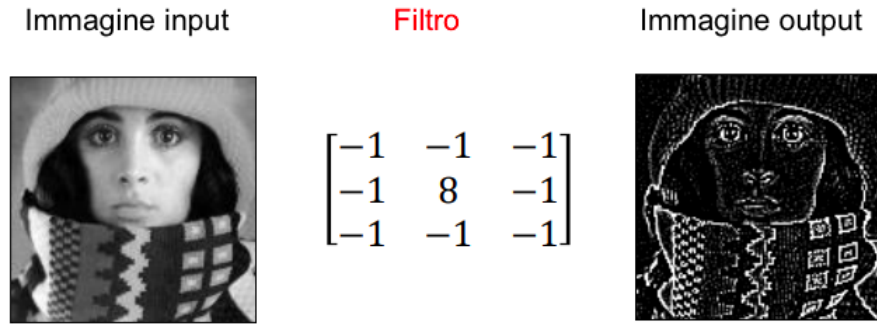


Figura 3.2: Esempio di un filtro e di un immagine "caratterizzata" dopo l'operazione di convoluzione

terizzata" dopo il passaggio del filtro (esempio in figura 3.2). Il numero di filtri che vengono applicati è un parametro da impostare. Solitamente nel primo livello di convoluzione si usano pochi filtri mentre il loro numero viene aumentato nei successivi livelli. Questo perchè per ogni filtro si genera un *output* e dato che la dimensione dell'*input* va progressivamente riducendosi livello dopo livello si preferisce applicare meno filtri ai livelli iniziali per mantenere contenuta la dimensione dell'*output* ad ogni livello. È anche per questo motivo che nei primi livelli di convoluzione si catturano dettagli più grossolani e meno precisi rispetto agli ultimi livelli di convoluzione dove si va ad effettuare un'analisi molto più nel dettaglio.

- *Livello Rectified Linear Units (ReLU)*: ha l'obiettivo di annullare eventuali valori negativi ottenuti dal prodotto scalare del livello precedente.
- *Livello di Pool (MaxPooling2D)*: permette di analizzare se la caratteristica che si sta esaminando nel livello attuale sia presente anche nel livello precedente. Inoltre semplifica l'immagine rendendola più grezza ma conservando tutte le caratteristiche. Consiste sempre nell'applicazione di un filtro che va a ridurre la dimensione della matrice che si è ricevuta in *input*.
- *Livello Full connection (FC)*: a questo livello si esegue la classificazione dopo i diversi livelli di convoluzione. Il modo in cui questo livello completamente connesso funziona è che guarda l'*output* del livello preceden-

te e determina quali caratteristiche sono maggiormente correlate a una particolare classe.

3.1.3 La struttura del modello ed esempio applicativo

L'idea che è a fondamento di questo lavoro, mirava ad alleggerire il carico di lavoro di un aerodinamico usando un modello che doveva essere in grado di etichettare le foto con una precisione superiore al 75% usando 23 tag.

All'inizio sono stati utilizzati dei modelli già allenati inclusi nelle librerie di Tensorflow [8] ma ci si è resi conto sin da subito che gli errori erano troppo alti in quanto gli algoritmi non riuscivano a distinguere tutte le minime differenze (nelle foto è necessario identificare dei punti molto specifici della macchina) delle varie appendici aerodinamiche da classificare.

Dopo un'attenta documentazione degli esempi presenti in letteratura, è iniziata l'elaborazione del programma partendo da poche semplici immagini di test. La scrittura generale del codice non ha richiesto molto tempo, il problema principale è stato il settaggio corretto dei valori delle reti di convoluzione, passaggio fondamentale se si vuole ottenere un'alta precisione.

la struttura del programma realizzato con la scuderia Haas di Formula 1 si compone di:

- *Interfaccia Grafica (GUI)*: creata con la libreria PySimpleGUI [40], permette all'utente di selezionare la cartella all'interno della quale ci sono tutte le foto da etichettare.
- *Lettura dei file*: le immagini vengono lette e conservate all'interno di vettori in modo da velocizzare il processo di lettura e scrittura di informazioni, per alleggerire il lavoro inoltre le immagini sono salvate nel vettore con una risoluzione di $300 * 200$ pixel.
- *Livelli di convoluzione*: questo modello è basato su 3 livelli di convoluzione con una maschera 3×3 e, dato il ridursi della dimensione dell'immagine all'avanzare dei livelli di convoluzione, un numero di filtri crescente per

ogni livello di convoluzione (128 nel primo livello, 256 nel secondo e 512 nel terzo) per andare a catturare sempre più dettagli all'aumentare dei livelli. Per ciascuno dei primi due livelli di convoluzione si è inoltre deciso di applicare un livello di *Pool* con dimensione 2 x 2 in modo tale da ridurre la dimensione dell'immagine e velocizzare il processo.

- *Allenamento*: dopo aver definito i livelli di analisi dell'immagine si è passati alla vera e propria fase di allenamento del programma. Non si useranno tutte le immagini dato che il 10% verrà conservato come insieme di test per analizzare eventuali problemi in fase di allenamento. In pratica il programma userà il 90% delle immagini già etichettate per imparare le caratteristiche diverse tra ogni classe, in seguito prova ad applicare ciò che ha imparato al restante 10% e lo confronta con le etichette già note. Si ha così una stima dell'accuratezza del programma.
- *Predizione per le nuove immagini date in input*: fino a questo punto il programma si è allenato ed ha usato solo delle immagini di test, a questo stadio del programma invece vengono fornite delle nuove immagini e il modello prova ad ipotizzare una possibile categorizzazione di quell'oggetto. Dato il largo numero di immagini simili tra di loro è stata implementata anche una funzione che analizza se l'immagine è simile alle precedenti e, nel caso lo sia, la salta in modo tale da avere una sola immagine con quell'etichetta.
- *Salvataggio dei dati*: è lo step finale di tutto il processo di etichettatura, salva le informazioni della parte della macchina all'interno dei metadati³ dell'immagine e in un file excel. Questo file excel sarà poi necessario per caricare le immagini su Sharepoint e renderle disponibili a tutto il dipartimento.

³I metadati sono informazioni aggiuntive, contenute all'interno di immagini digitali, che racchiudono dettagli tecnici dello scatto e identificativi dell'attrezzatura utilizzata.

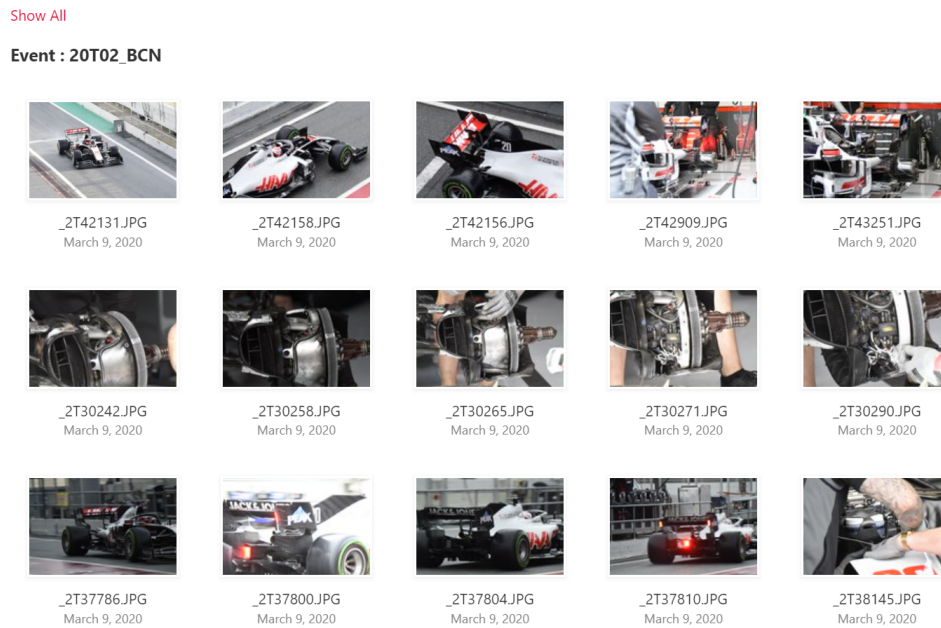


Figura 3.3: Filtri applicati: team Haas durante i Test di Barcellona del 2020

- *Analisi delle immagini:* a questo punto le immagini sono a completa disposizione dell'aerodinamico sulla piattaforma Sharepoint. Tramite dei filtri sull'area della macchina, il team e il gran premio si possono andare a visualizzare le varie soluzioni adottate velocemente e con pochi click. Nella figura 3.3 si può vedere un esempio di cosa si visualizza mettendo come filtro il team Haas e selezionando come evento i test di Barcellona nel 2020, mentre nella figura 3.4 si può vedere un esempio di visualizzazione se si filtra per tutte le immagini che raffigurano una monoposto vista da davanti. Si può quindi fare un'analisi veloce e dettagliata avendo subito tutte le informazioni necessarie a portata di *click*.

Il modello finale è stato diviso in due *script* principali: la prima parte che include tutto il processo di training e che dà come risultato un file .h5⁴. Quest'ultimo contiene tutte le informazioni necessarie alla seconda parte del programma per etichettare le foto. In questo modo si è diviso il lavoro in due con la parte di allenamento che non è necessario rieseguire a meno che non cambino le appendici aerodinamiche; inoltre lo spazio occupato dall'*output* dello *script*

⁴Un file .h5 è un file di dati salvato in Hierarchical Format Format (HDF), contiene array multidimensionali di dati scientifici

2020 > Front_view

Team : Ferrari



_1T28524.JPG
February 26, 2020



_1T45793.JPG
February 26, 2020



_1T46636.JPG
February 26, 2020



_1T46712.JPG
February 26, 2020



_1T46971.JPG
February 26, 2020



_1T46993.JPG
February 26, 2020



_1T33775.JPG
February 27, 2020



_1T34078.JPG
February 27, 2020



_1T35992.JPG
February 27, 2020



_1T36261.JPG
February 28, 2020



_1T36263.JPG
February 28, 2020



_1T36286.JPG
February 28, 2020



_1T38629.JPG
February 28, 2020



_1T48390.JPG
February 28, 2020



_1T19226.JPG
March 3, 2020



_1T19531.JPG
March 3, 2020



_1T19603.JPG
March 3, 2020



_1T19723.JPG
March 3, 2020

Team : Haas



_1T27500.JPG
February 26, 2020



_1T31669.JPG
February 27, 2020



_1T32041.JPG
February 27, 2020



_1T32064.JPG
February 27, 2020



_1T33755.JPG
February 27, 2020



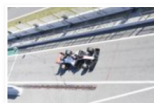
_1T28899.JPG
February 28, 2020



_1T38924.JPG



_1T48773.JPG



_1T20240.JPG



_1T29937.JPG



_1T29938.JPG



_1T29939.JPG

Figura 3.4: Filtri applicati: tutte le immagini di auto viste da davanti

di allenamento è nettamente inferiore rispetto a tutte le foto necessarie al programma per apprendere le informazioni indispensabili per distinguere le varie caratteristiche, di conseguenza l'applicazione può essere usata più facilmente da qualsiasi persona del dipartimento.

3.2 Problematiche affrontate

3.2.1 Overfitting

Il primo problema che si è dovuto affrontare è stato l'*overfitting*. Come visibile dalla figura 3.5 questo effetto si ha quando il modello durante l'allenamento ha delle prestazioni nettamente migliori mentre restituisce dei risultati molto meno precisi su nuove immagini. Questo è dovuto al fatto che l'algoritmo durante l'allenamento utilizza un set di immagini con troppa poca variabilità o troppo particolari, di conseguenza fallisce quando si vanno ad aggiungere dati diversi o immagini leggermente diverse. In poche parole il modello ricorda un largo numero di esempi invece di notare i dettagli.

Per risolvere il problema di adattamento eccessivo ai dati di allenamento, sono state adottate due soluzioni:

- *Aumento numero di immagini di allenamento*: dato l'elevato numero di classi e le poche immagini usate fino ad ora come allenamento, si è deciso di aumentare notevolmente il numero di foto classificate manualmente ed arrivare ad averne circa 5000, questo ha comportato un ritardo notevole nel completamento del programma ma ha garantito una migliore accuratezza.
- *Livelli di Dropout*: è un numero che va da 0 a 1 e permette di definire la percentuale di informazioni che viene esclusa casualmente passando da un livello a quello successivo. Consente quindi di concentrarsi solo su una parte delle caratteristiche identificate evitando che il modello si concentri sulla foto nel suo insieme.

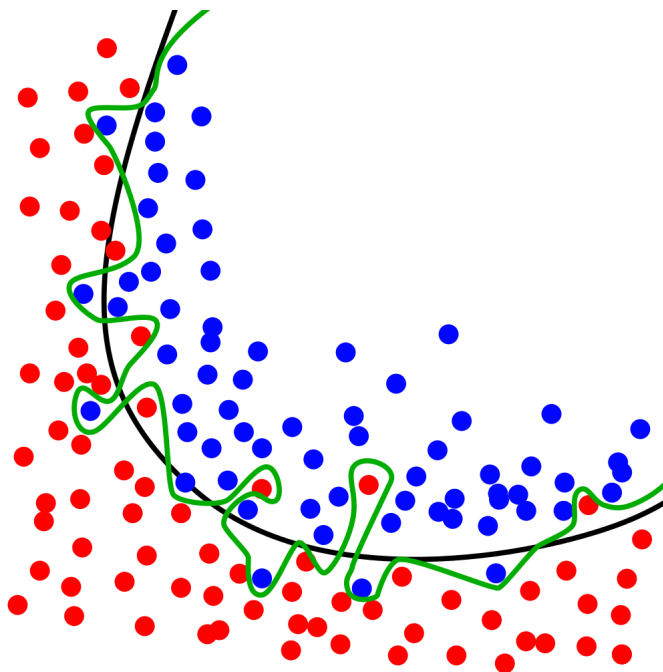


Figura 3.5: Overfitting: la linea verde rappresenta un modello con overfitting, la linea nera rappresenta un modello regolarizzato. La linea verde segue meglio i dati ma ne è troppo dipendente e si rischia di avere un errore molto più alto sui nuovi dati rispetto alla linea nera

3.2.2 Velocità di esecuzione

Il processo di training è estremamente lento (circa 7/8 ore) per cui ogni volta che veniva lanciato il modello era necessario diverso tempo per ottenere i risultati voluti. Si è deciso quindi di cercare un modo per separare tutto il processo di training (che era basato sempre sulle stesse immagini e che quindi era fisso) dal processo di etichettatura reale. Dopo una riscrittura parziale del problema si è riusciti a salvare all'interno di un file .h5 tutte le informazioni necessarie al programma per eseguire l'etichettatura. Questo passaggio ha permesso di risparmiare tantissimo tempo ad ogni avvio del programma dato che il training era già stato effettuato e doveva compiere solo l'identificazione dell'etichetta giusta per le nuove foto. Si è passati quindi da diverse ore per l'esecuzione dello script a pochi minuti.

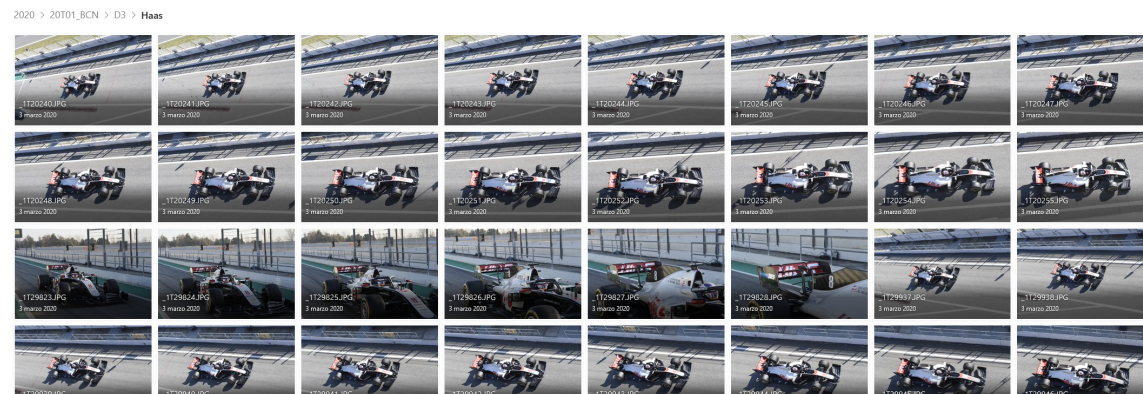


Figura 3.6: Esempio di visualizzazione con etichettatura effettuata a tutte le immagini

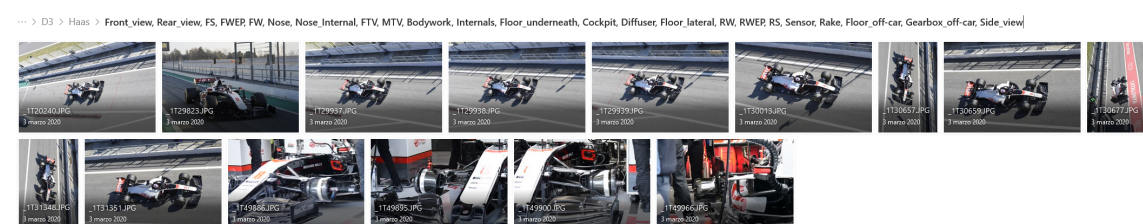


Figura 3.7: Esempio di visualizzazione con il filtro delle immagini uguali

3.2.3 Escludere le immagini uguali dal processo di etichettatura

Le foto effettuate dai fotografi di pista vengono catturate in rapida sequenza, quindi si hanno diverse immagini praticamente uguali tra di loro, si è deciso allora di rendere disponibile all'aerodinamico solo una foto dal set di queste immagini in modo tale da velocizzarne la ricerca. Questa richiesta ha comportato l'aggiunta di diverse funzioni all'interno dello script per analizzare la differenza tra le varie foto. Il modello è diventato leggermente più lento, ma questo ha permesso di poter gestire un minor numero di file. Questo è evidente confrontando le figure 3.6 e 3.7. Nel primo caso vengono mostrate tante immagini uguali (oltre 150 per ogni scuderia e per ogni singolo evento), mentre nel secondo caso il numero si riduce drasticamente permettendo una visualizzazione molto più veloce e completa.

3.2.4 Conservazione delle immagini e loro visualizzazione

Si sono prospettate diverse soluzioni sia costose che gratuite, i Digital Asset Management più costosi arrivavano anche ad oltre 10.000€. Dopo un'attenta analisi dei costi e dei vantaggi di tutte le soluzioni si è deciso di provare ad intraprendere la strada di Sharepoint (già utilizzato internamente). Questa decisione ha inciso sul codice già scritto perché richiedeva di ricevere le informazioni della categoria in formato testuale e non era possibile leggerle direttamente dai metadata delle immagini. È stato quindi implementato nello script del modello una funzione che scriveva l'etichetta predetta e il nome dell'immagine in un foglio Excel.

3.3 Confronto tra alcuni risultati ottenuti

Applicando le diverse tecniche e impostando i diversi parametri della rete convoluzionale si sono ottenuti risultati tra di loro molto diversi che hanno spinto nella scelta di una o dell'altra soluzione. Un elemento fondamentale per il miglioramento della qualità dell'*output* è stato, come già spiegato in precedenza, l'inserimento dei livelli di *dropout*. Nella tabella 3.1 sono riportati i risultati senza e con questi livelli. Di seguito vengono spiegati i valori indicati nelle immagini:

- 4636 è il numero di immagini su cui si allena l'algoritmo, il 5% di queste viene usata per effettuare un primo test di validazione;
- 524 è il numero di foto su cui prova a fare una previsione in base a quanto imparato durante l'allenamento;
- $s/sample$ è il tempo impiegato dal programma per processare ogni foto;
- $loss$ e val_loss sono i parametri che si deve cercare di minimizzare, il primo è relativo all'allenamento mentre il secondo al processo di validazione che si effettua su un sottoinsieme delle immagini di allenamento, più è alto questo valore più ci stiamo discostando dal valore esatto;

- *acc* e *val_acc* sono l'accuratezza del modello il cui massimo è 1 ed identifica una precisione del 100%;
- *Test score* e *Test accuracy* sono l'equivalente della funzione perdita e accuratezza del set di prova.

	senza dropout	con dropout
numero di immagini	4636	4636
tempo per processare una foto	0s 127us/sample	13s 3ms/sample
loss	0.0372	0.0320
acc	0.9896	0.9910
val_loss	1.0520	0.3213
val_acc	0.6935	0.8265
numero foto di prova	524	524
tempo per processare una foto di prova	0s 39us/sample	1s 963us/sample
loss	1.1146	0.3318
acc	0.6896	0.8115
Test score	1.1146	0.3318
Test accuracy	0.6896	0.8115

Tabella 3.1: Risultati senza e con i livelli di dropout

Analizzando la tabella 3.1 è visibile fin da subito come, senza i livelli di *dropout*, il modello ha degli ottimi risultati durante l'allenamento con una precisione intorno al 98%, mentre le prestazioni sono nettamente inferiori sia durante la validazione che la prova su 524 immagini già etichettate rimanendo sempre al di sotto del 70%.

Confrontando questi risultati con quelli ottenuti dopo aver inserito i livelli di *dropout*, si nota come sia aumentata nettamente la precisione durante la fase di prova e di validazione di oltre il 12% raggiungendo un'accuratezza superiore all'80%. Nella fase di allenamento i parametri rimangono pressochè costanti intorno al 99%. Un altro parametro che è stato influenzato dall'inserimento del *dropout* è il tempo di processamento, passato da frazioni di secondo per ogni immagine ad oltre un secondo. La causa è da attribuire al fatto che il modello

si concentra molto di più sui dettagli e di conseguenza ha bisogno di più tempo per processare tutte le informazioni necessarie.

Capitolo 4

Conclusioni

In conclusione si può dire che questo lavoro ha permesso di:

- catalogare in tempi ridotti un ingente quantitativo di foto;
- assicurare agli aerodinamici un set di foto di alta qualità;
- trovare foto in brevissimo tempo tramite delle ricerche molto dettagliate;
- semplificare, alleggerire e velocizzare il lavoro degli aerodinamici.

I risultati ottenuti sono stati:

- precisione dell'82% (con gli script preallenati si era raggiunta al massimo un'accuratezza del 60%)
- tempo per processare nuove foto di circa 4 minuti ogni 100 immagini (notevolmente inferiore rispetto al tempo richiesto dal programma iniziale che includeva ogni volta il processo di allenamento)

4.1 Miglioramenti nel programma elaborato

Il *machine learning* è un potentissimo mezzo da utilizzare in ambiti come quello della Formula 1 dove i dati da gestire sono ingenti e ogni decimo di secondo guadagnato è fondamentale. I campi di applicazione possono andare dall'analisi di casi di CFD fino all'analisi dei dati di pista. Vista la centralità dell'apprendimento automatico sarebbe utile, in futuro, un corso di studi dove si integrino

parzialmente l'informatica e l'aerodinamica così da avere aerodinamici con adeguate competenze informatiche che possano sfruttare al meglio le potenzialità dell'intelligenza artificiale.

4.1.1 Utilizzo di un software diverso da Sharepoint

Sharepoint purtroppo fino ad ora è un po' limitato per la visualizzazione delle immagini a causa di una memoria di salvataggio non elevata e di un'articolata gestione dell'*upload* delle nuove foto. Una possibilità è l'utilizzo di un interfaccia web programmata usando *Django Web Framework* [41], basato sempre su linguaggio Python. È molto veloce ed adattabile a tutte le esigenze.

4.1.2 Migliorare la precisione del modello

La precisione attuale del modello raggiunge circa l'80%, di conseguenza un'immagine su 5 ha l'etichetta sbagliata. Una soluzione per migliorare questo aspetto è di inserire le immagini già categorizzate all'interno dell'allenamento in modo da ingrandire l'insieme di immagini date come allenamento e di conseguenza migliorarne la precisione. Il problema è che le immagini da inserire tra quelle di allenamento devono subire un check da parte di un utente in modo da evitare che troppe immagini catalogate erroneamente finiscano all'interno del set di allenamento

4.2 Il machine learning e la CFD

Un campo dove ci sono tantissimi dati da analizzare è la CFD. Gli aerodinamici ricevono tantissime immagini come risultato del post pro e a volte può essere difficile tenere traccia di tutti i flussi. Potrebbe essere utile un modello che, partendo dalle foto che gli aerodinamici vedono come output, tracci questi flussi e identifichi i vortici principali intorno all'automobile. Questo permetterebbe di avere un output più immediato oltre ad avere una diversa visualizzazione dei dati.

Bibliografia

- [1] Mitchell. *Machine Learning*. 1997.
- [2] Alan M Turing. Computing machinery and intelligence. In *Parsing the Turing test*, pages 23–65. Springer, 2009.
- [3] Marvin Minsky. *Society of mind*. Simon and Schuster, 1988.
- [4] Marvin Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.
- [5] Frank Rosenblatt. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Lab Inc Buffalo NY, 1961.
- [6] Stanley Cohen. The evolution of machine learning: past, present, and future. In *Artificial Intelligence and Deep Learning in Pathology*, pages 1–12. Elsevier, 2021.
- [7] Sandra Vieira, Walter Hugo Lopez Pinaya, and Andrea Mechelli. Introduction to machine learning. In *Machine Learning*, pages 1–20. Elsevier, 2020.
- [8] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore,

- Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [10] Microsoft’s Technology and Research division. The microsoft cognitive toolkit, 2021.
- [11] Jia Li and James Z Wang. Real-time computerized annotation of pictures. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):985–1002, 2008.
- [12] Yong-Ku Kim and Kyoung-Sae Na. Application of machine learning classification for structural brain mri in mood disorders: Critical review from a clinical perspective. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, 80:71–80, 2018.
- [13] Claas Ahlrichs and Michael Lawo. Parkinson’s disease motor symptoms in machine learning: A review. *arXiv preprint arXiv:1312.3825*, 2013.
- [14] Vanessa Aguiar-Pulido, José A Seoane, Juan R Rabuñal, Julián Dorado, Alejandro Pazos, and Cristian R Munteanu. Machine learning techniques for single nucleotide polymorphism—disease classification models in schizophrenia. *Molecules*, 15(7):4875–4889, 2010.

- [15] Jefferson Tales Oliva, Huei Diana Lee, Newton Spolaôr, Claudio Saddy Rodrigues Coy, João José Fagundes, Maria de Lourdes Setsuko Ayrizono, and Feng Chung Wu. Ontology-based process for unstructured medical report mapping. In *Machine Learning in Bio-Signal Analysis and Diagnostic Imaging*, pages 1–18. Elsevier, 2019.
- [16] Abhijit Bandyopadhyay, D Ravi Shankar, Arun S Tigadi, and Aatif Jamshed. An advanced machine learning technique for the analysis of retina fundus images. *Materials Today: Proceedings*, 2020.
- [17] Jesper Sören Dramsch. 70 years of machine learning in geoscience in review. *Advances in Geophysics*, 2020.
- [18] TD Jobe, E Vital-Brazil, M Khait, et al. Geological feature prediction using image-based machine learning. *Petrophysics*, 59(06):750–760, 2018.
- [19] Bastian Bohn, Jochen Garcke, Rodrigo Iza-Teran, Alexander Paprotny, Benjamin Peherstorfer, Ulf Schepsmeier, and Clemens-August Thole. Analysis of car crash simulation data with nonlinear machine learning methods. *Procedia Computer Science*, 18:621–630, 2013.
- [20] Jack Stilgoe. Machine learning, social learning and the governance of self-driving cars. *Social studies of science*, 48(1):25–56, 2018.
- [21] Chaoran Zhou, Hongfei Jia, Jingxin Gao, Lili Yang, Yixiong Feng, and Guangdong Tian. Travel mode detection method based on big smartphone global positioning system tracking data. *Advances in Mechanical Engineering*, 9(6):1687814017708134, 2017.
- [22] Yawen Hu, Yunpeng Wang, Junfeng Zhang, and Haiyang Yu. Correlation degree analysis of arterial adjacent intersections for coordinated control subunit partition. *Advances in Mechanical Engineering*, 10(1):1687814017748748, 2018.
- [23] Inc. Amazon.com. Aws powers, fl insights, 2019.

- [24] Luigi Accardi Guido Altarelli Pietro Dominici Mauro La Forgia Giuseppe Tedone, Adriano Alippi. Dizionario di scienze fisiche della enciclopedia italiana, 1996.
- [25] Giulio Costanzi Modesto Panetti. Enciclopedia italiana, treccani, 1929.
- [26] Joseph Katz. Aerodynamics of race cars. *Annu. Rev. Fluid Mech.*, 38:27–63, 2006.
- [27] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.
- [28] Francesco Capizzano. *Sviluppo e applicazione di un metodo numerico per lo studio di flussi turbolenti intorno a geometrie navali*. PhD thesis, Università degli Studi di Napoli Federico II, Ottobre 2002.
- [29] Brian Edward Launder and Dudley Brian Spalding. The numerical computation of turbulent flows. In *Numerical prediction of flow, heat transfer, turbulence and combustion*, pages 96–116. Elsevier, 1983.
- [30] Willem Toet. Aerodynamics and aerodynamic research in formula 1. *The Aeronautical Journal*, 117(1187):1–26, 2013.
- [31] Triya Nanalal Vadgama, Mr Arpit Patel, and Dr Dipali Thakkar. Design of formula one racing car. *International Journal of Engineering Research & Technology (ISSN: 2278-0181)*, 4(04):702–712, 2015.
- [32] Ben Agathangelou and Mike Gascoyne. Aerodynamic design considerations of a formula 1 racing car. Technical report, SAE Technical Paper, 1998.
- [33] Umberto Ravelli and Marco Savini. Aerodynamic simulation of a 2017 f1 car with open-source cfd code. *J. Traffic Transp. Eng*, 6, 2018.
- [34] Alex Guerrero and Robert Castilla. Aerodynamic study of the wake effects on a formula 1 car. *Energies*, 13(19):5183, 2020.

- [35] Cloudinary. Cloudinary, 2021.
- [36] Daminion Software. Daminion, 2020.
- [37] Razuna. Razuna, 2021.
- [38] Microsoft. Sharepoint, 2019.
- [39] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [40] The PySimpleGUI Organization. Pysimplegui, 2020.
- [41] Django Software Foundation. Django, 2019.