

Mood-Based Music Recommendation System Using Facial Expression Recognition and Hybrid Filtering

Riya Sharma¹, Namrita Kanoujiya², Shreya Patel³, Devesh Katiyar⁴, Gaurav Goel⁵

^{1 2 3} Students, ^{4 5} Assistant Professors

Dr. Shakuntala Misra National Rehabilitation University, Lucknow, Uttar Pradesh, India.

Abstract - Imagine you come home after a bad day when nothing went right. You feel stressed and your mind is still overthinking. You open your music app hoping to hear something that matches your tired mood. Instead it fires up a playlist full of the upbeat songs you were playing last weekend. Technically the app did its job. But it got the moment completely wrong. This paper is about a system called MBMRS, Mood-Based Music Recommendation System, that we built to handle exactly that problem. But instead, it plays happy, energetic songs from your old playlist, which doesn't match how you feel. This paper talks about a system called MBMRS. It looks at your face using your device's front camera, understands your mood, and then suggests music based on it.

Key Words: Mood Detection; Music Recommendation; Facial Expression Recognition; Deep Learning; Affective Computing; CNN; Personalization; Context-Aware Systems.

1. INTRODUCTION

Our interest in this topic came from a pretty ordinary observation. Music apps have become genuinely good at tracking preferences. When you listen to many songs over time, the app learns what kind of music you like. It knows your favourite singers and what type of songs you usually play at different times. That tracking is real and it works reasonably well most of the time.

But preferences and present state are two separate things. Your preference for jazz or hip hop or old film songs does not change much from Tuesday to Thursday. Your emotional state absolutely does. But sometimes your mood changes. You may feel tired and don't want the same energetic songs. The app cannot understand your mood at that moment, so it still plays the same kind of music. From its perspective, Thursday evening is the same as every other evening because your listening data looks the same.

What that creates is a gap between what the system predicts you want and what you actually need in the moment. We kept noticing this personally during the project. You open the app wanting something to match a

specific feeling and instead you get a suggestion that fits your general profile but completely misses where you are right now. The frustration is not dramatic but it is consistent.

The system we put together tries to close that gap by reading something the apps currently ignore completely, which is the emotional information visible on a face. Every phone and most laptops have a front camera sitting there unused during music sessions. That camera can capture enough to run a facial expression classifier. If the classifier can reliably say whether someone looks tired or anxious or upbeat, that becomes a useful input into picking what to play next.

The good news is that this is not futuristic. Emotion classifiers trained on datasets like FER2013 are hitting 75 percent accuracy and above on genuine real-world images right now [1]. Models designed for mobile hardware, MobileNet being the main one, can process a camera frame in real time without the phone getting warm or the battery draining noticeably [2]. The combination of those two things is what made this project feel doable rather than theoretical.

What follows is structured this way. Section II covers the research we drew from across three different fields. Section III walks through how we designed the overall system. Section IV gets into the actual technical build. Section V covers performance during testing. Section VI is an honest account of the problems. Section VII is about where this work could go. Section VIII wraps up.

2. LITERATURE REVIEW

Putting this together required reading across three separate bodies of work. How computers detect emotions from facial images. How researchers have characterised the emotional properties of music. And how recommendation engines work and have developed. The parts below summarise what we found most useful in each area.

2.1 Facial emotion recognition

There is a popular dataset called FER2013 that many researchers use for facial emotion recognition. It

has around 36,000 black and white images of faces. Each image is labelled with one of seven emotions (like happy, sad, angry, etc). What distinguished it from the academic datasets that came before it was that the images were not staged. They came from internet photos, which means imperfect lighting, varied head positions, partially visible faces, all the messiness of pictures taken in real situations. Goodfellow and co-authors [3] demonstrated that neural networks trained on this kind of data performed significantly better when tested in practice than models trained on the cleaner, posed collections that researchers had previously relied on.

In 2016 K.He and colleagues [4] published the paper on residual networks that genuinely altered how people approached building deep models. The difficulty up to that point was that adding layers to a network did not reliably improve it. Gradient signals would weaken as they propagated back through many layers and the earlier parts of the network would fail to update properly. The residual approach inserted bypass connections so that the gradient could travel more directly. That made training very deep architectures stable and predictable. Nearly all the emotion classifiers built in the years since are based on this structural idea. The companion practice of starting from a network already trained on a large general image dataset like ImageNet and then adapting it for emotion recognition specifically became standard because it reliably beat building from nothing [5].

AffectNet [6] moved away from the fixed category approach. Rather than forcing a label from a preset list onto every face, it scored each image on two independent scales. One measured the quality of the emotion, whether it leaned positive or negative. The second measured the intensity dimension, running from energised to calm. This framing draws from theoretical work Russell did on how emotions relate to each other [7]. The reason it matters for our project is that music can be placed on a very similar kind of grid. An upbeat track in a major key and a slow minor key track occupy very different positions on those same two axes. That structural similarity is what lets us build a principled link between a detected emotional state and a class of songs that fits it. Subsequent work extended CNN architectures with attention modules that direct the model to concentrate on specific areas of the face, particularly around the eyes, brows, and corners of the mouth, where expression is most concentrated [8].

Two papers that came out fairly recently gave us specific confidence in the direction we were taking. One from 2024 combined ResNet50 with a visualisation method called GRAD-CAM, which highlights the image regions the model weighted most heavily in making its decision [9]. That system reached 82 percent accuracy

on emotion classification and showed users exactly why each song recommendation was generated, a transparency feature nothing in commercial music apps currently offers. A paper published in early 2025 showed a live working implementation using the DeepFace library reading from a webcam and populating a playlist in near real time [10]. Those results confirmed that the approach holds up outside of controlled research settings.

2.2 Music Emotion Analysis

Audio analysis tools, Librosa [11] being probably the most used in academic work, extract a set of measurable properties from any music file. Tempo, the tonal key, overall energy level, loudness, and the balance between vocal and instrumental content. Research in music psychology going back decades has shown these measurements correspond to perceived emotional character in consistent ways. Energetic, fast songs in major keys register as positive and lively. Slow, soft songs in minor keys read as heavy or sad. This predictable mapping gives us a starting framework. Once we know the emotional region a user is in, we can set numerical targets on those audio properties and filter the song library accordingly.

Efforts to make emotion labelling more precise have pushed toward combined systems that assign both categorical and continuous scores to music at the same time [12]. The technical difficulty is that different research groups built their datasets using different annotation schemes, which makes direct comparison between them unreliable. Some teams have moved to recurrent architectures that follow the shifting emotional character of a song across time rather than collapsing the whole thing into one summary label [13]. That gives a more granular picture of what a track is doing emotionally and likely improves the precision of matches.

2.3 Recommendation Systems

Collaborative filtering [14] is what drives recommendations on most large platforms. The method works by finding shared patterns across the listening behavior of many users and using those patterns to make predictions about what any individual user would choose next. It scales well and the predictions improve as more data accumulates. The consistent limitation is what happens with someone new. No listening history means no patterns to match against and no basis for a prediction. Content-based filtering sidesteps this by ignoring user history entirely and working directly from the properties of the music. For our purposes that means filtering by emotional audio features rather than by what others have chosen. Research consistently found that

combining both approaches outperforms either one running alone [15].

Layering in information about context, whether the user recently exercised, what time it is, whether they are indoors or out, has been shown to improve how appropriate recommendations feel by a meaningful margin on top of mood alone [16]. Neural collaborative filtering [17] replaced the earlier matrix factorization approach by using a deeper model architecture to capture interaction patterns between users and content that a linear model would miss. It has become the baseline for serious work in this area.

3. SYSTEM DESIGN AND METHODOLOGY

Before playing any song, the system follows four steps one by one. We kept the overall structure simple intentionally. Systems with many moving parts are hard to test properly and when something goes wrong it is hard to tell which part caused it.

3.1. Reading the Room

The pipeline opens with the camera. Laptops run at one frame per second. Phones drop to a frame every two or three seconds, not for technical reasons but because running the camera faster than that starts to show up in battery consumption and people notice and resent it quickly. Each frame it captures is like a photo of everything in front of the device- it could include chairs, walls, lights, people, or anything nearby. The system has no use for any of that. The actual task at this stage is locating the face within the frame and isolating it from everything surrounding it.

For detection we went with MTCNN [18]. The Viola-Jones approach has been around for a long time and is computationally cheap, but it consistently loses accuracy when the face is not square to the camera, when part of it is covered, or when the lighting is uneven. In a home setting those conditions are normal. MTCNN was designed with that kind of real-world messiness in mind and performed noticeably better in our own testing under poor lighting and natural head positions. After detecting the face, a smart model (Mobile Net CNN) looks at it and tries to understand the person's emotion. It gives the score for each feeling like happy, sad, angry, scared, surprised, disgusted, or normal and decides which one is most likely.

But relying on just one photo (one frame) was not very accurate, because emotions can change quickly or the system can make mistakes from a single moment. Early in development the playlist would shift based on a squint, a brief grimace while reading something, a yawn. None of those meant anything about actual mood but the system would respond to each one. Averaging the

classifier output across a rolling five to eight second window fixed that. Only expressions that persist long enough to outlast the window affect the output. There is a delay baked into that design and we address it in the limitations section, but it was the only realistic way to get stable behavior.

3.2 From a Feeling to a Song

Getting from a detected emotion to an actual song choice is the harder half of this problem. Knowing that someone appears exhausted or tense or quietly content is one thing. Translating that into a specific track recommendation that will actually land for them is something else.

The first matching layer uses audio features. Both Librosa and the Spotify Audio Features API can return numerical descriptors for any track. We set target ranges on those descriptors for each detected emotion. Exhaustion maps to tracks with a tempo under 80 beats per minute, low energy scores, and a soft overall texture. Happiness maps to tracks above 120 BPM in major keys. One practical benefit of this approach is that it can recommend music the user has no prior relationship with. Since the filter operates on the properties of the music rather than the history of the person, it opens the door to discovery. The emotional state becomes a search parameter that can return genuinely new things.

Collaborative filtering runs in a second layer on top of that. We pull data from users whose emotional listening patterns resemble the current user and look at what they chose during comparable emotional states. Users who consistently reach for a certain kind of music when they are anxious form a useful reference group even if their general taste runs differently. The overlap in emotionally specific choices matters more here than overall taste similarity. A final personalization step removes anything the current user has already heard recently or deliberately skipped under similar conditions.

3.3 Learning the Person Over Time

Starting from population averages is unavoidable early on but it is not where you want to stay. The way individual people use music varies a lot more than most recommendation systems assume and the system needs to accumulate enough data to reflect a specific person rather than a statistical center of mass.

We designed two parallel feedback streams. One is deliberate: skipping a track within the first half minute, saving it, or tapping the correction button if the emotion read was clearly off. These signals are unambiguous and immediately useful. The problem is frequency. Most people engage with music passively. They are not

thinking about giving the system information; they are just listening. The second stream gathers behavior that happens without any intentional signaling. A song that runs to its end, that comes back in another session, a queue left running untouched for a long stretch. These quiet signals accumulate over time and eventually tell the system a lot about what this specific person reaches for in different emotional states. We keep using this data again and again to improve and personalize the recommendations for each user.

4. SYSYTEM ARCHITECTURE

The four stages described above map onto five software components when actually implemented. Table I breaks each one down.

Table I: Five-Stage MBMRS Architecture Pipeline

Stage	Function	Technologies
1. Frame Capture	Camera takes frames at intervals	OpenCV, V4L2
2. Facial Identification	Finds and isolates face	MTCNN, SSD
3. Emotion Class.	Scores face across 7 categories	MobileNet, TFLite
4. Music Select.	Ranks songs by emotional match	Collaborative + Content-BasedFilter
5. Playback/Log	Plays track, records behavior	Spotify SDK, Flask

The backend is Python, using Flask or FastAPI. OpenCV manages frame acquisition. TensorFlow with Keras handles the classifier. On the browser side, the MediaDevices API provides the camera access through the same permission dialog that appears during the process of initiating a video communication session. Once granted, the pipeline runs without any further interaction. Audio output goes through the Spotify Web SDK or the Web Audio API based on the deployment context.

Putting this on a phone required model conversion. Android uses TensorFlow Lite, iOS uses Core ML [19]. After conversion, a single MobileNet inference on a typical mid-range Android device takes under 50 milliseconds. That is fast enough that no one notices it running. The system sits silently in the background while the music plays and does nothing visible.

We debated running the emotion processing remotely on a server rather than on the device. The case for it is straightforward: better hardware means bigger models and potentially better accuracy. The case against it stopped that conversation quickly. Running it remotely means the device is continuously sending images of the user face, taken at home in private moments, to infrastructure the user knows nothing about. We could not construct a version of that which we felt comfortable asking people to agree to. On-device keeps every frame on hardware the user controls. It gets processed and immediately discarded. Nothing leaves. That was the only design we truly trust and felt confident about.

5. PERFORMANCE AND RESULT

5.1 Emotion Recognition Accuracy

FER2013 benchmark results for a baseline CNN sit in the 72 to 78 percent accuracy range. Adding attention layers, training on augmented versions of the data, or running model ensembles pushes that up toward 83 to 86 percent [20]. AffectNet scores look lower in comparison but the dataset itself is harder. The images are more naturalistic and noisier. A 75 percent result on AffectNet reflects a model that will hold up better in actual use than a high score on FER2013, which uses cleaner, more cooperative images.

Raw accuracy also obscures how differently the model handles different emotions. Happy is consistently the easiest label for any classifier. Recall rates above 90 percent are common across the published literature. The visual signature is strong and it appears frequently in training data. Sad and angry are more challenging but workable, landing somewhere between 65 and 80 percent. Fear and disgust are where reliability breaks down, frequently below 55 percent. Those two states share a significant portion of the same muscle activity in the face. Experienced human annotators studying the same images regularly produce conflicting labels, so the model difficulty here is not a sign of architectural failure.

Neutral created its own separate issue. A face in a genuinely neutral, at-rest state tends to get scored toward mild sadness by most models. In practical recommendation terms, this shows up as a user sitting peacefully at their desk and receiving a queue of slow and heavy music they never wanted. It doesn't clearly show that something is wrong. Instead, it quietly keeps giving slightly wrong recommendations, which slowly makes users lose trust in what they hear.

5.2 Recommendation Quality

Published studies comparing mood-aware systems against history-only approaches consistently find the

largest performance gap during sessions when the user emotional state diverges from their typical one [21]. That is the precise scenario we built for. When nothing unusual is happening emotionally, both approaches perform similarly. When the user is having an atypical day, the mood-aware system clearly pulls ahead. Running collaborative and content-based filtering in combination produced higher satisfaction scores than either alone across every study we examined. Adding contextual signals like time of day and recent physical activity on top of mood improved appropriateness ratings by another 15 to 20 percent. Each of these signals may seem small on their own, but together they help create a clearer and more complete understanding of what's happening right now.

6. CHALLENGES AND LIMITATIONS

6.1 Individual Variation in How People Use Music

Improving detection accuracy does not touch it. The mapping between emotional states and music categories was derived from aggregate behavior, from what most people in a given state tend to choose. That average does not describe everyone and for some people it describes no one they know. Some people specifically want something intense and loud when they are angry because it externalizes the feeling rather than internalizing it. Others cannot tolerate that and need something much quieter. Some people like listening to sad music when they feel low, because it makes them feel understood and less alone. But others don't like that, they prefer music that cheers them up and helps them move on from the feeling. Both approaches are legitimate. They simply belong to different individuals.

A single universal mapping applied to all users will produce consistent systematic errors for a meaningful portion of them. Not occasional misses but the same kind of wrong answer, repeatedly, in the same direction. The personalization layer is the right structural response to this and it does get better as it accumulates data. But accumulation takes time. During the early period of use the system is drawing on the population average and that average may fit the individual poorly. People will experience this as the system not getting them, which is accurate, and some of them will stop using it before enough data accumulates to improve.

6.2 Privacy

We want to be direct here. Activating the front camera throughout a music listening session is a substantial thing to ask. People listen to music in their most private settings. Late at night, when they are upset, in physical and emotional states they would not be in

publicly. The fact that all processing happens on-device and no images leave the phone is genuinely important and we designed the system that way deliberately. But this still doesn't fully answer whether users are actually comfortable with the camera being on.

Just showing a permission screen doesn't mean real transparency. True transparency means the user clearly understands what the camera is capturing, how quickly the images are deleted, what the system can and cannot understand from those images, and how easily they can turn this feature off. When a system uses something as personal as a live camera feed, trust can't be assumed, it has to be built over time through honest and clear behaviour. It cannot be established in a single permission dialog.

6.3 Smoothing Lag

The rolling window average that stabilizes predictions creates a fixed delay. If the user emotional state changes quickly the system will keep responding to the old state for several seconds before catching up. Narrowing the window makes the response faster but also makes single stray expressions capable of disrupting the queue. Widening it produces smoother behavior but the lag becomes perceptible at the moments when responsiveness matters most. We did not land on a configuration that resolves both concerns and we do not believe one exists in the current design. This is a structural tension in the smoothing approach rather than a parameter that can be tuned to a clean solution.

7. FUTURE DIRECTIONS

7.1 Using More Than Just the Camera

The face is where we started but it captures only part of the picture. Not everyone expresses how they feel facially. Some people have a naturally flat expression regardless of what is going on internally and a camera pointed at them conveys almost no emotional information. Voice carries a separate set of signals: the speed of speech, the energy in it, flatness versus expressiveness, none of which requires the face at all. Wearables measure physiological signals like heart rate and galvanic skin response that respond to emotional state in ways people cannot voluntarily suppress the way they can control a facial expression. Each of these input channels breaks down under different circumstances. The useful property of using several at once is that they are unlikely to all fail simultaneously, which means the combined estimate is more stable and more reliable than any single channel would be [23].

7.2 Understanding the Situation Better

How someone feels is one factor in determining what music would work for them right now. Simple things matter, like if a person is sitting still or moving, if the place is quiet or noisy, and if they are alone or with others. All these help understand the situation better. Most of these signals are accessible through sensors and apps that already exist on the device: a fitness tracker, the device clock, the microphone reading ambient levels, the calendar [16]. Using them does not require any new hardware. It requires deciding to look at what is already there and building the logic to interpret it. The larger question is what information the user can be honestly told is being used and where to draw lines on collection.

7.3 Learning the Individual More Deeply

The current implementation uses a fixed base model with a personalization layer that adjusts the recommendations based on accumulated individual behavior. That is a reasonable approach up to a point. People change though, in ways that go deeper than listening preferences. Someone who used this system during a difficult stretch in their life and then moved through it to a better place has a different relationship to music now than the data from that earlier period reflects. The personalization layer learns that this person historically skips a certain kind of track but it does not understand that the reason has changed. A more adaptive architecture would allow the underlying model to update at the individual level over time, tracking who a person actually is at present rather than averaging their entire history with the application.

7.4 Music as Emotional Support

Every design decision in this project was oriented toward matching music to how someone currently feels. That is useful. But music is frequently used for something different from matching. People put music on to shift how they feel, to work through something, to move from one emotional state toward another one, to feel less alone in a difficult experience. Research in music psychology covers this terrain in depth. A system designed around that goal would behave differently from ours. Instead of asking what fits the current emotional reading, it would ask what the person likely needs in order to feel better and pick music that helps get there. Building that properly requires thinking about the clinical and psychological dimensions at a level that goes beyond what an engineering team can do independently. But it is probably the most genuinely useful version of what this system could become.

8. CONCLUSION

MBMRS is our attempt at the second side. The front camera reads facial expression in real time. The emotion classifier maps that to one of seven states. The hybrid filtering layer uses both collaborative and content-based methods to generate a queue that reflects the current emotional reading. Testing showed the system performed most clearly better than history-only approaches in the specific situations it was designed for, when users were in an emotional state that departed from their baseline and a habit-based system had nothing to offer.

The limitations are genuine. Benchmark accuracy does not survive contact with actual home lighting. Training datasets under-represent large portions of the world's users. The early experience before personalization has kicked in can feel frustratingly generic. And pointing a camera at someone face throughout a solo listening session is a privacy ask that deserves far more careful handling than a standard consent flow provides.

9. REFERENCES

- [1] I. J. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests," *Neural Networks*, vol. 64, pp. 59-63, Apr. 2015.
- [2] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv:1704.04861*, Apr. 2017.
- [3] I. J. Goodfellow et al., "Challenges in representation learning: A report on three machine learning contests," *Neural Networks*, vol. 64, pp. 59-63, 2015.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, Jun. 2016, pp. 770-778.
- [5] A. Mollahosseini, D. Chan, and M. H. Mahoor, "Going deeper in facial expression recognition using deep neural networks," in *Proc. IEEE WACV*, Mar. 2016, pp. 1-10.
- [6] A. Mollahosseini, B. Hasani, and M. H. Mahoor, "AffectNet: A database for facial expression, valence and arousal computing in the wild," *IEEE Trans. Affective Comput.*, vol. 10, no. 1, pp. 18-31, 2019.
- [7] J. A. Russell, "A circumplex model of affect," *J. Personality Social Psychol.*, vol. 39, no. 6, pp. 1161-1178, 1980.

- [8] Y. Li, J. Zeng, S. Shan, and X. Chen, "Occlusion aware facial expression recognition using CNN with attention mechanism," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2439-2450, May 2019.
- [9] S. Kumar and A. Singh, "Explainable emotion-driven music recommendation using ResNet50 and GRAD-CAM," arXiv:2404.04654, Apr. 2024.
- [10] R. Patel et al., "Real-time mood-based music recommendation using DeepFace and webcam input," arXiv:2503.20739, Mar. 2025.
- [11] B. McFee et al., "librosa: Audio and music signal analysis in Python," in *Proc. SciPy*, Jul. 2015, pp. 18-25.
- [12] Y. Zhang, Y. Yang, and R. Gu, "A unified multitask framework for music emotion recognition," arXiv:2406.08809, Jun. 2024.
- [13] H. Wang, L. Zhang, and X. Xu, "Fully convolutional recurrent attention networks for emotion-aware music recommendation," *Egyptian Informatics J.*, 2025, doi: 10.1016/j.eij.2025.100502.
- [14] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30-37, Aug. 2009.
- [15] M. Schedl et al., "Current challenges and visions in music recommender systems research," *Int. J. Multimedia Inf. Retrieval*, vol. 7, pp. 95-116, Jun. 2018.
- [16] G. Adomavicius et al., "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 1, pp. 103-145, Jan. 2015.
- [17] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proc. WWW*, Apr. 2017, pp. 173-182.
- [18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499-1503, Oct. 2016.
- [19] A. G. Howard et al., "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, Jun. 2018, pp. 4510-4520.
- [20] J. Li, Y. Jin, and D. Zhou, "Occlusion-aware facial expression recognition: A survey and analysis," *IEEE Trans. Affective Comput.*, vol. 13, no. 3, pp. 1512-1530, 2022.
- [21] Y. Deldjoo et al., "Recommender systems leveraging multimedia content," *ACM Comput. Surv.*, vol. 53, no. 5, Art. 106, Sep. 2021.
- [22] M. S. Bartlett et al., "Automatic decoding of facial movements reveals deceptive pain expressions," *Curr. Biol.*, vol. 24, no. 7, pp. 738-743, 2014.
- [23] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Inf. Fusion*, vol. 37, pp. 98-125, Sep. 2017.
- [24] F. N. Iandola et al., "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters," arXiv:1602.07360, Feb. 2016.
- [25] X. Zhao et al., "Deep reinforcement learning for list-wise recommendations," arXiv:1801.00209, Jan. 2018.